



Universidad
Carlos III de Madrid

PROYECTO FIN DE CARRERA
Ingeniería Técnica de Informática de Gestión

TÍTULO

MODELO DE META-METADATOS XBRL

Autor: Jorge Valencia Martínez

Tutores: Ignacio J. Santos Forner y Elena Castro Galán

Leganés, 27 de Septiembre de 2011

Agradecimientos

Agradezco a Ignacio y a Elena, tutores de este proyecto, por todo el esfuerzo dedicado, por todo lo aprendido y por todo lo disfrutado. Igualmente, por la confianza depositada en mi, y por la paciencia que ha tenido durante el desarrollo del trabajo.

Agradezco de la misma forma el apoyo dado por mi familia, Maxi, Raquel y Belen, que nunca desistieron en su insistencia para que realizara este proyecto, y a Irene, que aguantó mis noches en vela trabajando para su conclusión.

Índice General

1.	INTRODUCCIÓN AL META-METAMODELO DE DATOS XBRL.....	16
1.1	Metamodelo y XBRL	17
1.2	Objetivos del proyecto	17
1.3	Fases del Desarrollo y medios empleados	17
2.	INTRODUCCIÓN A XML.....	19
2.1	Orígenes de XML.....	19
2.2	Document Type Definition	22
2.3	Documentos Válidos y bien formados.....	23
2.4	Elementos y atributos.....	24
2.5	XML SCHEMA.....	25
2.6	Vinculación XML	30
2.7	Tecnologías XML.....	32
3.	INTRODUCCIÓN A XBRL.....	33
3.1	Contexto histórico	33
3.2	Orígenes.....	34
3.3	Nacimiento y Evolución de XBRL	35
3.4	¿Qué es XBRL?	37
3.5	Lenguaje XBRL	39
3.6	Arquitectura XBRL.....	41
3.7	Proyectos XBRL en el mundo.....	46
4.	META-METAMODELO XBRL.....	48
4.1	Objetos XML	49
4.2	Tipos de Datos XML	56
4.3	Objetos XBRL	59
4.3.1	Atributos XBRL.....	59
4.3.2	Grupos de Atributos XBRL	67
4.3.3	Tipos Simples XBRL	75
4.3.4	Tipos Complejos XBRL.....	86
4.3.5	Elementos XBRL.....	127
5.	HERRAMIENTAS MODELADO XML-UML.....	187

5.1 Herramientas	187
6. CONCLUSIÓN Y TRABAJO FUTURO	191
7. BIBLIOGRAFÍA	192

Índice de Figuras

Figura 1. Meta-metamodelo dentro de XBRL.....	16
Figura 2. Optimización XBRL.....	38
Figura 3. Taxonomía y roles.....	40
Figura 4. Bases de un documento Instancia.....	42
Figura 5. Roles de la capa conceptual	43
Figura 6. Roles de la capa de relaciones.....	44
Figura 7. Ejemplo de estructura de DTS	45
Figura 8. Capa de extensiones.....	46
Figura 9. Niveles de modelado XBRL I	48
Figura 10. Niveles de modelado XBRL II	49
Figura 11. Elemento “context”	156
Figura 12. Ejemplo de modelo multidimensional para el elemento “context”.....	157
Figura 13. Ejemplo de elemento “hypercube”	179
Figura 14. Ejemplo de elemento “explicitMember”.....	185
Figura 15. Ejemplo de “typeMember”	186

Índice de Objetos XBRL

Objeto “simpleType”	49
Objeto “complexType”	50
Objeto “element”	50
Objeto “attribute”	50
Objeto “simpleContent”	51
Objeto “complexContent”	51
Objeto “attributeGroup”	52
Objeto “substitutionGroup”	52
Objeto “restriction”	53
Objeto “extensión”	53
Objeto “anyAttribute”	54
Objeto “choice”	54
Objeto “sequence”	55
Tipo de datos “Name”	56
Tipo de datos “NCName”	56
Tipo de datos “NMTOKEN”	57
Tipo de datos “NMTOKENS”	57
Tipo de datos “anyURI”	57
Tipo de datos “processContent”	57
Tipo de datos “QName”	58
Atributo xlink:actuate.....	59
Atributo xlink:arcrole.....	59
Atributo xlink:from	60
Atributo xlink:href	60
Atributo xlink:label	60
Atributo xlink:role.....	61
Atributo xlink:show	61
Atributo xlink:title	61
Atributo xlink:to.....	62
Atributo xlink:type.....	62
Atributo xbrldt:closed.....	63

Atributo xbrldt:contextElement	63
Atributo xbrldt:typeDomainRef.....	63
Atributo xbrldt:usable	64
Atributo xbrldt:targetRole	64
Atributo xbrli:balance.....	64
Atributo xbrli:periodType	65
Grupo xlink:titleType	67
Grupo xlink:simpleType	67
Grupo xlink:resourceType	68
Grupo xlink:locatorType	68
Grupo xlink:extendedType	69
Grupo xlink:arcType.....	69
Grupo xbrli:factAttrs.....	70
Grupo xbrli:itemAttrs	71
Grupo xbrli:essentialNumericItemAttrs	71
Grupo xbrli:nonNumericItemAttrs	72
Grupo xbrli:numericItemAttrs	72
Grupo xbrli:tupleAttrs	73
Tipo xlink:nonEmptyURI	75
Tipo xbrli:dateUnion.....	78
Tipo xbrli:decimalsType.....	79
Tipo xbrli:monetary	80
Tipo xbrli:nonZeroDecimal	81
Tipo xbrli:precisionType	82
Tipo xbrli:pure	83
Tipo xbrli:shares	83
Tipo xbrldt:contextElementType.....	84
Tipo xl:titleType	86
Tipo xl:simpleType.....	86
Tipo xl:resourceType	88
Tipo xl:locatorType	89
Tipo xl:extendedType	90
Tipo xl:arcType	92
Tipo xl:documentationType.....	94

Tipo xbrli:anyURIItemType	95
Tipo xbrli:NCNameItemType	96
Tipo xbrli:base64BinaryItemType	96
Tipo xbrli:booleanItemType	97
Tipo xbrli:byteItemType	98
Tipo xbrli:dateItemType	98
Tipo xbrli:dateTimeItemType	99
Tipo xbrli:decimalItemType	100
Tipo xbrli:doubleItemType	100
Tipo xbrli:durationItemType	101
Tipo xbrli:floatItemType	102
Tipo xbrli:fractionItemType	102
Tipo xbrli:gDayItemType	103
Tipo xbrli:gMonthDayItemType	104
Tipo xbrli:gMonthItemType	105
Tipo xbrli:gYearItemType	105
Tipo xbrli:gYearMonthItemType	106
Tipo xbrli:hexBinaryItemType	107
Tipo xbrli:integerItemType	107
Tipo xbrli:intItemType	108
Tipo xbrli:longItemType	109
Tipo xbrli:measuresItemType	109
Tipo xbrli:monetaryItemType	110
Tipo xbrli:negativeIntegerItemType	111
Tipo xbrli:nonNegativeIntegerItemType	111
Tipo xbrli:nonPositiveIntegerItemType	112
Tipo xbrli:normalizedStringItemType	113
Tipo xbrli:positiveIntegerItemType	113
Tipo xbrli:pureItemType	114
Tipo xbrli:QNameItemType	115
Tipo xbrli:sharesItemType	115
Tipo xbrli:shortItemType	116
Tipo xbrli:stringItemType	117
Tipo xbrli:timeItemType	117

Tipo xbrli:tokenItemType	118
Tipo xbrli:unsignedByteItemType.....	119
Tipo xbrli:unsignedIntItemType	119
Tipo xbrli:unsignedLongItemType	120
Tipo xbrli:unsignedShortItemType	121
Tipo xbrli:contextEntityType	121
Tipo xbrli:contextPeriodType	123
Tipo xbrli:contextScenarioType	125
Elemento link:arcroleRef	127
Elemento link:arcroleType.....	128
Elemento link:calculationArc.....	130
Elemento link:calculationLink.....	131
Elemento link:definition	132
Elemento link:definitionArc.....	133
Elemento link:definitionLink.....	134
Elemento link:documentation.....	135
Elemento link:footnote	136
Elemento link:footnoteArc	137
Elemento link:footnoteLink.....	138
Elemento link:label.....	139
Elemento link:labelArc.....	140
Elemento link:labelLink	141
Elemento link:linkbase	142
Elemento link:linkbaseRef	143
Elemento link:loc	145
Elemento link:part	145
Elemento link:presentationArc.....	146
Elemento link:presentationLink	147
Elemento link:reference	148
Elemento link:referenceArc.....	149
Elemento link:referenceLink	149
Elemento link:roleRef	151
Elemento link:roleType	152
Elemento link:schemaRef.....	153

Elemento link:usedOn	154
Elemento xbrli:context	155
Elemento xbrli:denominator	158
Elemento xbrli:numerator	158
Elemento xbrli:divide	158
Elemento xbrli:item	159
Elemento xbrli:measure	159
Elemento xbrli:segment	161
Elemento xbrli:tuple	162
Elemento xbrli:unit	163
Elemento ref:Appendix	164
Elemento ref:Article	164
Elemento ref:Chapter	165
Elemento ref:Clause	165
Elemento ref:Example	166
Elemento ref:Exhibit	166
Elemento ref:Footnote	167
Elemento ref:IssueDate	168
Elemento ref:Name	168
Elemento ref:Note	169
Elemento ref:Number	169
Elemento ref:Page	170
Elemento ref:Paragraph	171
Elemento ref:Publisher	171
Elemento ref:Section	172
Elemento ref:Sentence	172
Elemento ref:Subclause	173
Elemento ref:Subparagraph	173
Elemento ref:Subsection	174
Elemento ref:URI	175
Elemento ref:URIDate	175
Elemento xbrldt:dimensionItem	177
Elemento xbrldt:hypercubeItem	178
Elemento xbrli:xbrl	180

Elemento xbrldi:explicitMember	184
Elemento xbrldi:typedMember	186

1. INTRODUCCIÓN AL META-METAMODELO DE DATOS XBRL

Este proyecto final de carrera pretende realizar una primera aproximación al modelo de meta-metadatos que define el lenguaje de reporting XBRL. Es un primer paso en el que se estudiarán los conceptos utilizados en su definición, analizando el propio metamodelo sobre el que se asienta. La definición de este meta-metamodelo consistirá en definir los objetos existentes, sus relaciones, e incluso los enlaces con el propio lenguaje XML, ya que el lenguaje XBRL está basado en XML. Este proyecto es una parte de un proyecto mayor que consiste en definir su modelo conceptual y poder mapear este modelo a otros modelos conceptuales. En la figura 1 podemos ver un esquema global, y la parte concreta que estamos analizando.

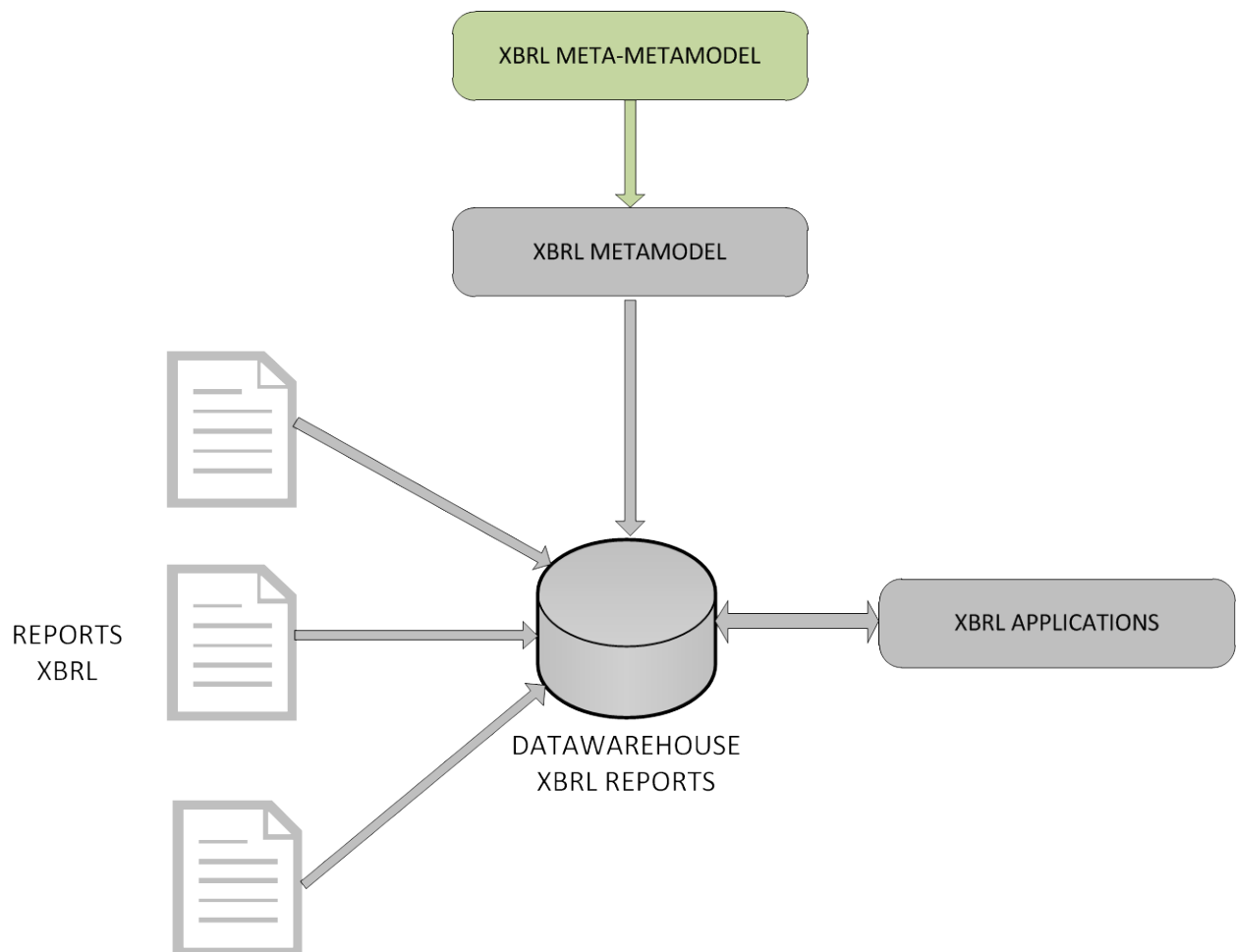


Figura 1. Meta-metamodelo dentro de XBRL

1.1 Metamodelo y XBRL

Primeramente, deberemos entender que un metamodelo, concretamente, de un lenguaje, es un modelo de información para definir el propio lenguaje. Dicho con otras palabras, es el análisis, construcción y desarrollo de las reglas y objetos que definen un determinado lenguaje. Este lenguaje sobre el que hemos realizado el estudio es XBRL, lenguaje de informes de negocio extensible (eXtensible Business Reporting Language).

XBRL es un lenguaje surgido en el entorno financiero que es utilizado para la presentación de informes contables, y que permite la comunicación, de forma estandarizada, entre empresas u organismos [3]. XBRL permite el intercambio de información financiera entre entidades, así como su tratamiento y almacenamiento para su posterior gestión. Esta tecnología, basada en XML, está basada en el uso de taxonomías comunes, y compartidas, de conceptos financieros. De esta forma se trabaja sobre un sistema conceptual común y estandarizado. A la hora de interpretarlo, estas taxonomías permiten interpretar los hechos de forma clara y concisa. Además, XBRL es validado en origen, por lo que se puede decir que es sintácticamente correcto. Representa información de negocio, la cual es multidimensional, por lo que los informes financieros podrían almacenarse en un DataWareHouse [11-25].

Al ser XBRL un lenguaje basado en XML, las referencias a este último lenguaje serán habituales en este proyecto, incluyendo una introducción que permita tener una base de conocimiento para entender mejor el objeto de estudio de este trabajo.

1.2 Objetivos del proyecto

Cualquier metamodelo es el punto de partida de un estudio y desarrollo mayor. La definición de un meta-metamodelo de XBRL supone entender el lenguaje, poner los cimientos para las construcciones de objetos que puedan tratar XBRL, leerlo, procesarlo, interpretarlo, poder comunicarse con él. En la actualidad, XBRL va extendiendo sus redes, dada su utilidad, es importante conseguir que su progreso no cese, y que cada vez sean más los organismos, nacionales e internacionales, que hagan uso de él. El estudio de XBRL, junto con su meta-metamodelo, tiene como objetivo final el colaborar en ese desarrollo, en facilitar su extensión y abaratar los costes de su evolución.

1.3 Fases del Desarrollo y medios empleados

En este proyecto se ha trabajado en varias fases, iniciándose el mismo sobre un estudio introductorio del lenguaje XBRL, con el fin de realizar una labor de aprendizaje primario, que nos permita tener unas nociones básicas para la posterior profundización. A continuación, era necesario conocer XML, origen del lenguaje objeto del estudio, para después ahondar en el propio XBRL, estudiando su metamodelo y los cimientos sobre los que se asienta. Para finalizar, se ha realizado un pequeño estudio de herramientas que podrían

utilizarse para trabajar con modelos UML y archivos XML, que permitan gestionar el paso de uno a otro, y facilitar el trabajo de estudio y gestión de ambos.

Respecto a los medios empleados, en este caso, además de la bibliografía y las referencias indicadas, se ha empleado la herramienta Microsoft Visio 2010, para el modelo de datos y Microsoft Visual Studio 2010 para el análisis de las taxonomías estudiadas.

2. INTRODUCCIÓN A XML

Para poder entender XBRL, es necesario conocer previamente el lenguaje sobre el que está basado. A continuación realizaremos una introducción de lenguaje XML, desde sus orígenes, pasando por los principales objetos que le dan forma, como los DTD's ("Document Type Defintion"), los elementos y atributos, estableciendo la diferencia entre documentos válidos y documentos bien formados, analizando el concepto de XML Schema y el de vinculación, para terminar viendo la evolución hasta nuestros días del propio lenguaje XML.

2.1 Orígenes de XML

Como hemos mencionado con anterioridad, XML no es otra cosa que un estándar definido para ser utilizado en la creación de nuevos lenguajes estructurados, y que a su vez permita ser leído e interpretado por diversas plataformas. Un documento XML es un medio estructurado para almacenar información [16].

Para hablar de los orígenes de XML, debemos remontarnos a los años sesenta, cuando se perseguía la idea de estructurar los documentos de forma organizada, para facilitar su intercambio y manipulación. IBM, a raíz de todo esto, creó GML ("Generalized Markup Language" - Lenguaje de Marcado Generalizado). GML se utilizó para producir libros, informes y otros documentos a partir de un solo conjunto de archivos fuente. Algo más tarde, y también IBM, definió y adoptó el estándar SGML ("Standard Generalized Markup Language" – Lenguaje Estandar de Marcado Generalizado) para el intercambio de información (ISO-8879). SGML se utilizó para dar formato y mantener documentos legales en IBM. En 1986 se declara estándar ISO (ISO-8879). SGML se basa en el marcado y etiquetado de los documentos, sin definir cuáles deben ser esas etiquetas. Gracias a SGML la estructura de un documento permite ser definida en base a la relación lógica de sus partes. Dicha estructura será validada por lo que se conoce como DTD ("Document Type Definition"). Por definición, un lenguaje de marcado se puede entender como un sistema de descripción de información. SGML era el pionero en unificar los criterios para describir esa información [36].

SGML, como hemos dicho, no especifica qué etiquetas utilizar, si no que da las normas para definir esas etiquetas. Por ello es más correcto hablar de metalenguaje, ofreciendo de esta manera una mayor flexibilidad en la definición de lenguajes de marcas, y a su vez, potencia en el intercambio y conservación de documentos y recursos estructurados.

Un documento SGML, consta de tres partes: declaración SGML, declaración del tipo de documento (DTD) e instancia del documento [1].

La declaración SGML indica qué puede y qué no puede un usuario incluir en el contenido de un documento SGML. Nos dirá los caracteres, los delimitadores y el resto de características que proceda definir.

La DTD es donde se define la estructura del documento. Se definirán los elementos que son necesarios para la elaboración de uno o varios documentos SGML similares. Contiene reglas de nombrado, significados, descripción de contenidos, uso de los propios elementos...Sin la DTD, no es posible interpretar el documento SGML.

Por último, está la instancia, que es el contenido, la información, los datos. Veámoslo con un ejemplo sencillo basado en la DTD anterior:

Por lo tanto, como podemos observar, la información (instancia) es inseparable de la definición de su estructura (DTD), y a su vez, ésta, es definida de acuerdo a la propia declaración SGML. Es una secuencia fácilmente deducible por los sistemas informáticos, que validarán el documento [6].

Un ejemplo de lenguaje basado en el estándar SGML, no es otro que HTML. Surgido a raíz de la evolución de “World Wide Web”, en 1989, fueron dos investigadores del Laboratorio Europeo de Física de Partículas (CERN), Tim Berners-Lee y Ander Berglund, los que definieron un lenguaje basado en etiquetas para marcar documentos técnicos con el fin de compartirlos a través de Internet. Por lo tanto, HTML es una definición de un tipo de documento utilizado en la Web.

HTML supuso una revolución, provocando la creación de innumerables tecnologías para añadir interactividad a la Web. Algunas de ellas, como Java, se mantienen hoy en día, y siguen su curso, mientras que otras han ido perdido presencia progresivamente. HTML fue creado para presentar información estática. Dentro de toda esta evolución, se intentó dar a HTML la opción de presentar información dinámica, lo que iba en contra de su esencia. Además, se pretendía ampliar el espectro de etiquetas que utiliza HTML (en un principio, fijo), algo que SGML permitía.

En febrero de 1998 nace la primera especificación de XML (XML 1.0, “eXtensible Markup Language” – Lenguaje de Marcado Extensible) como consecuencia de la evolución de HTML apoyada en SGML.

La expansión de la Web y la necesidad de intercambio de archivos de gran volumen, hizo que se buscara un lenguaje que pudiera sustituir a HTML, ya que éste era poco potente para las necesidades que iban surgiendo. En 1996 el W3C decidió introducir el poder y flexibilidad de SGML en la Web. Se pensó que SGML podría aportar sus tres principales características a HTML: extensibilidad, estructura y validación.

En 1998 se consiguió cerrar la primera especificación, y se definió como un “Sistema para definir, validar y compartir formatos de documentos en la Web”. Había nacido la primera especificación de XML (“eXtensible Markup Language” – Lenguaje de Marcas Extendido). La diferencia principal entre HTML y XML es que el primero se centra en la presentación de los datos, mientras que en el segundo lo principal son los datos, la información. Por ello, XML no es sustituto de HTML, son dos conceptos distintos, y algunas de sus diferencias básicas son [31]:

- XML es independiente de navegadores y sistemas de objetos.
- No requiere DTD
- XML posee punteros a la estructura de datos.
- XML es más estricto formalmente.

Uno de los usos que hemos comentado de XML, es el intercambio de ficheros en la Web. Sin embargo, no es exclusivo de este entorno. Se usa como formato de intercambio entre diferentes plataformas, sean cuales sean éstas, y ahí radica su principal propiedad, es un metalenguaje estandarizado y fácilmente manejable.

XML consiste en unas reglas que definen formatos de contenidos. Estos ficheros son sencillos de construir y de leer. Las etiquetas definidas en XML sirven para delimitar los datos, permitiendo que sea la aplicación la que los interprete.

Cabe mencionar los principios sobre los que surgió XML, y que han permitido que a día de hoy esté tan extendido que cualquier nueva tecnología tenga en cuenta el manejo de este tipo de documentos:

- XML debe poder ser utilizado en Internet
- Debe permitir su uso por multitud de aplicaciones
- Debe ser compatible con SGML
- Las aplicaciones que lo utilicen deberían ser sencillas
- No debe ser extensible
- Los documentos XML deben ser legibles e interpretables por las personas
- Se debe basar en un diseño rápido
- Debe estar normalizado
- Los documentos XML deben ser construidos de manera sencilla
- La concisión de las marcas no debe ser relevante

Gracias a estos conceptos iniciales, XML da una flexibilidad enorme a la interpretación de los datos, dejándola en manos del destinatario. A su vez, es más sencillo que SGML, en todos sus aspectos, desde la creación de documentos, hasta la creación de programas que lean XML. La información será reutilizable. El uso de XML se ha extendido en tres entornos principalmente:

- La web
- Comunicación de aplicaciones
- Configuración de programas

A raíz de todo esto, han surgido nuevas aplicaciones asociadas a XML:

- Centrados en la presentación de los datos (DOM)
- Buscadores (XML/XSL)
- Creación de DTD para la definición de una gramática, o web semántica (XTM, “XML Topics Maps”)

Pero no todo son ventajas, también existe alguna crítica, basado principalmente en el nivel de detalle y complejidad que pueden alcanzar sus extensiones. El mapeo con otros lenguajes de programación o bases de datos puede resultar complicado, haciéndose más patente esa dificultad cuando se manejan datos muy estructurados (que no era un objetivo en sus orígenes).

A modo de conclusión, podemos decir que XML es flexible y extensible, puesto que no son necesarias versiones de adaptación. Es más sencillo de manejar, y su usabilidad abarca todos los ámbitos de las nuevas tecnologías. Permite el intercambio entre diferentes plataformas. Facilita las búsquedas debido a su estructuración del contenido. Y por último, es fácilmente exportable a cualquier otro tipo de formato.

2.2 Document Type Definition

La DTD (“Document Type Definition”, o Definición de Tipo de Documento) conforma la base de los documentos válidos, puesto que establece la gramática de un vocabulario XML, que, a su vez, determina la estructura de los documentos XML. Una DTD sirve para llevar a cabo la validación de documentos, que es una parte importante del desarrollo de contenidos en XML. Sin las especificidades de los elementos y sus relaciones, es imposible validar un documento [16].

En XML, la estructura de un documento viene determinada por la declaración de tipo de documento, que aparece en el prólogo del propio documento XML. En la declaración de tipo de documento, las declaraciones de marcado establecen la gramática de un documento o clase de documento. La DTD es la gramática que hay en la declaración de tipo de documento, o a la que ésta hace referencia. En otras palabras, la declaración de tipo de documento establece el punto inicial para describir la estructura de un documento, y la DTD es quien en realidad la describe.

La declaración de tipo de documento puede llevar a cabo una de estas funciones:

- Contener directamente declaraciones de marcado en el prólogo del documento, conocidas como subconjuntos de la DTD interna o DTD local.
- Hacer referencia a declaraciones externas de marcado que se conocen como subconjunto de la DTD externa.

De esta forma, la DTD se divide en dos partes, una interna y otra externa. Cuando se hace referencia a la DTD de un documento, se hace referencia a todo el conjunto. La división permite una mayor flexibilidad, ya que la DTD externa se utiliza para definir la estructura general del documento de una clase de documentos, y la DTD interna para describir una estructura específica de un documento en concreto. Los procesadores XML dan prioridad a la DTD interna, por lo que es posible “sobreescribir” declaraciones de la DTD externa definiéndolas de nuevo en la interna.

En una DTD se pueden definir las siguientes informaciones:

- Elementos
- Atributos
- Entidades
- Notaciones

Todas ellas forman la estructura lógica y física de una clase de documentos, o “tipo de documento”.

Como hemos comentado anteriormente, se puede definir la estructura de un documento XML mediante una DTD interna, una DTD externa, o por medio de ambas. Si se va a utilizar una de ellas, deberá especificarse en la declaración del documento. La sintaxis sería:

`<!DOCTYPE ElementoRaíz SYSTEM DTDExterna [DTDInterna]>`

DTDExterna es el “Uniform Resource Identifier” (Identificador Uniforme de Recursos) de un archivo que contiene la DTD externa. La DTD se declara entre corchetes. Podemos observar que el otro punto importante de la declaración es el elemento raíz, que identifica precisamente el elemento raíz de la clase de documento (la declaración de tipo de documento debe aparecer después de la declaración XML, pero antes del primer

elemento de un documento. A continuación vamos a ver un ejemplo concreto de declaración de tipo de documento:

```
<!DOCTYPE asignatura SYSTEM "Asignaturas.dtd" [
```

```
<!ELEMENT profesor (#PCDATA)>]>
```

- Todos los elementos de este tipo tendrán el elemento 'asignatura'
- Se hace referencia a una DTD externa llamada Asignaturas.dtd
- Se declara un elemento profesor que forma parte de la DTD interna.

En la declaración XML se puede especificar también si el documento es autónomo o no. Es decir, si se apoya en fuentes de información externas (como pudieran ser DTDs) o no. El atributo standalone (Yes/No) es el que lo indica. Por defecto, si no se señala nada, se toma el valor Yes:

```
<?xml versión "1.0" standalone="no"?>
```

Algunas de las razones para decidir utilizar una DTD externa serían:

- Crear documentos que deben ser válidos.
- Crear varios documentos de la misma clase.
- Usar una DTD ya existente.
- Crear documentos concisos.

También existen casos en los que tal vez es más óptimo utilizar una DTD interna

- No es relevante que el documento sea válido.
- Se crea un único documento.
- Se pretende minimizar el coste asociado a un documento.

En definitiva, dependerá de la situación en que nos encontremos para optar por una opción u otra.

2.3 Documentos Válidos y bien formados

Hemos visto que un documento XML se divide en dos partes, la definición del esqueleto (DTD) y los datos (archivo XML). En función de estas dos partes podemos diferenciar dos tipos de documentos: "Válidos" y "Bien formados".

Un documento es válido cuando cumple los siguientes requisitos [16]:

- El documento debe estar bien formado.
- El nombre del elemento raíz debe ser el mismo que el nombre de la declaración de tipo de documento.
- El documento debe tener una DTD que declare todos los elementos, atributos y entidades que se utilicen en el documento (interno, externo o ambos).
- El documento debe adherirse a la gramática que establezca la DTD.

Para que un documento sea considerado bien formado, debe cumplir:

- Tener un elemento de documento en el que se encuentren todos los demás documentos (elemento raíz).
- Todo elemento tendrá una etiqueta de inicio y otra de cierre, o una sola vacía.
- Todos los elementos estarán correctamente anidados.
- Todos los valores de atributos irán entre comillas (dobles o simples).
- Todas las entidades que se utilicen deberán declararse en la DTD (externa o interna).

Por tanto, y como podemos deducir, todos los documentos DTD válidos están bien formados, pero no todos los documentos bien contruidos son documentos válidos.

2.4 Elementos y atributos

El concepto principal de una estructura XML es lo que se conoce como “elemento”. Un elemento posee identidad propia, y permite definir la estructura lógica del documento. Un elemento puede contener otros elementos, lo que implica una jerarquización entre ellos. Como hemos explicado anteriormente, todo fichero XML consta de un único elemento raíz, que engloba a todos los demás. Además, todo documento XML comienza por un elemento opcional, denominado “declaración” o “prolog”, donde se especifica la versión utilizada, así como la codificación de caracteres. Posteriormente, deberá aparecer el elemento raíz, que será el que contenga el resto de objetos del documento. Véase un ejemplo:

```
<?xml version=" 1.0 " encoding=" UTF-8 " standalone= " yes ">
```

```
<pelicula>
```

```
<titulo>El Señor de los Anillos</titulo>
```

```
<director>Peter Jakson</director>
```

```
</pelicula>
```

Vemos que por un lado están las marcas que indican el principio y el fin del elemento, y por otro, los datos contenidos en dichas marcas. La primera línea indica la versión, la codificación y si lleva asociado un DTD.

Además de los elementos, también existen los atributos, que son características de la información definida por un elemento, es decir, propiedades de los elementos, y que, al contrario que estos últimos, no contienen información compleja ni permiten relaciones jerárquicas:

```
<pelicula género="ciencia ficción" categoría="superproducción">El Señor de los Anillos</pelicula>
```

Vamos a enumerar alguna de las ventajas del uso de atributos frente al uso de elementos [16]:

- Los atributos pueden limitarse frente a una lista predefinida de valores enumerados.
- Los atributos pueden tener valores predeterminados.
- Los atributos tienen tipos de datos (aunque limitados).
- Los atributos son muy concisos.

- Lo atributos son más fáciles de analizar sintácticamente que los elementos.

Sin embargo, también existen desventajas:

- Los atributos no son recomendables para cadenas de texto largas.
- Los atributos no pueden contener información anidada.
- Los espacios en blanco no pueden omitirse en un valor de atributo.

Por todo ello, se recomienda usar un atributo cuando la información requiere un valor predeterminado o fijo, o datos que son embebidos de un elemento existente. Además, los atributos suelen ocupar menos bytes que los elementos.

2.5 XML SCHEMA

Para validar un documento XML, siempre se ha necesitado el uso de las DTD, heredadas de SGML. A pesar de su utilidad, mostraban ciertas deficiencias, como pudieran ser la sintaxis críptica, la falta de introducción de datos, el modelo de contenido cerrado y la falta de soporte para los espacios de nombres. XML Schema es una tecnología creada por Microsoft, mediante una implementación de un subconjunto de una nota del W3C dirigida a mejorar las DTD (1998). XML Schema nace como un vocabulario XML [21].

La definición dada por W3C para un esquema XML es el conjunto de reglas que sirve para forzar la estructura y la articulación del conjunto de documentos XML. En 1998, W3C recibió una nota acerca de la utilidad DCD (“Document Content Description”, Descripción del Contenido del Documento), basada en el vocabulario RDF (“Resource Description Framework”, Estructura de Descripción de Recursos). RDF es un vocabulario XML diseñado para proporcionar una base para procesar metadatos, permitiendo que las aplicaciones web representen y compartan información en un formato interpretable por la máquina. DCD es un vocabulario RDF para describir estructuras de documentos.

Hemos hablado brevemente del lenguaje XML, de que consta de dos partes, la definición del propio documento, y el contenido. Además, hemos dicho que XML no precisa necesariamente de un DTD. El concepto que sustituye al DTD, es lo que se denomina XML Schema (aunque esto no quiere decir que no puedan utilizarse DTDs).

XML Schema cumple la función de definir la estructura de los documentos XML que estén asignados a dicho esquema, así como los tipos de datos válidos para cada elemento y atributo. Permite a su vez restringir y documentar el significado de las diferentes partes que forman un archivo XML, y precisar el uso e interrelaciones que lo constituyen. Es, por tanto, una especie de vocabulario de documentos XML. Igualmente, la posibilidad de definir el contenido, permite intercambiar información sin necesidad de demasiadas comprobaciones.

Cualquier aplicación que haga uso de XML podrá utilizar el estándar XML Schema para expresar restricciones estructurales y valores aplicables a las instancias de una clase de documentos.

Algunas de las aplicaciones prácticas de XML Schema son [16]:

- Distribución de información a través de servicios de publicación, relacionando imágenes, cabeceras o elementos de noticias.
- Procesamiento de transacciones en el comercio electrónico estableciendo relaciones de mercado entre las partes.
- Control de supervisión y adquisición de datos mediante el uso de mensajes de validación y control en el intercambio de información entre nodos de una red.
- Control de la auditoría y edición de documentos tradicionales mediante la creación de plantillas, para facilitar el trabajo del usuario.
- Ayuda en el uso y optimización de consultas.
- Transferencia abierta y uniforme de datos entre aplicaciones y bases de datos creando esquemas que facilite el propio intercambio.
- Intercambio de metadatos que faciliten la interoperabilidad de diseños de bases de datos, gestores de bases de datos, interfaces de usuario...

Según W3C, las especificaciones de XML Schema se rigen por una serie de 'principios' que dicen:

- El lenguaje XML Schema debe:
 - 1) Proporcionar mayor expresividad que las DTDs de XML.
 - 2) Está basado en lenguaje XML.
 - 3) Ser autodescriptivo.
 - 4) Poder ser usado por cualquier aplicación que utilice XML.
 - 5) Permiten especificar los tipos de datos.
 - 6) Poder utilizarse directamente en internet.
 - 7) Ser optimizado e interoperable.
 - 8) Ser simple para utilizarse en implementaciones.
 - 9) Poder coordinarse con otros estándares definidos en W3C, como HTML o RDF Schema.
- Además, XML Schema debe poder definir:
 - 1) Mecanismos para restringir la estructura y el contenido de los documentos.
 - 2) Herencia entre los diferentes objetos que lo formen (elementos, atributos y tipos de datos).
 - 3) Mecanismos para referenciar URI's ("Universal Resource Identifier") de forma estándar.
 - 4) Sistemas para incluir documentación.
 - 5) Mecanismos para proporcionar descripciones y restricciones de aplicación específica.
 - 6) La dirección de la evolución de los esquemas.
 - 7) La integración entre esquemas estructurados y datos primitivos.
- Igualmente, en lo que respecta a los tipos de datos, se aclara que deben [39]:
 - 1) Proporcionar el mecanismo de tratamiento de todo tipo de datos, entre los que se encuentran los de acceso a web y los de bases de datos.
 - 2) Permitir la representación léxica de la información.
 - 3) Permitir la definición de nuevos tipos de datos basados en los ya predefinidos.

Este tipo de ficheros se basan en una estructura modular, permitiendo la reutilización de los módulos ya definidos. Además, entre dichos módulos se establecen relaciones jerárquicas, lo que da lugar a dos tipos de objetos dentro de un esquema XML: tipos simples y tipos complejos.

Tipos simples son aquellos que no tienen ni elementos hijos, ni atributos. Entre ellos, están los tipos predefinidos en XML (“double”, “string”, “boolean”...), los tipos lista (enumerados) y las uniones de tipos predefinidos.

Por ende, son tipos complejos aquellos que tienen elementos hijos y/o atributos.

Los elementos que se pueden definir en un XML-Schema son, principalmente:

ELEMENTO	DESCRIPCIÓN
Schema	Identifica el comienzo de una declaración de “Schema”
Datatype	Especifica el tipo de dato de un ElementType o AttributeType
ElementType	Define un tipo de elemento para usarlo como elemento del “Schema”
Element	Hace referencia a un elemento declarado anteriormente mediante una etiqueta “elementType”
Group	Organiza contenidos en un grupo para especificar una secuencia
AttributeType	Define un tipo de atributo para usarse dentro de un elemento del “Schema”
Attribute	Hace referencia a un atributo declarado anteriormente mediante una etiqueta
Description	Suministra información adicional sobre un ElementType o AttributeType

Elemento Schema

El elemento Schema, como hemos dicho, es el nodo raíz del documento. Contiene el resto de objetos. Posee dos atributos, que definen el nombre y el espacio de nombres del esquema. Son, respectivamente, “name” y “xmlns”.

El atributo xmlns tiene gran importancia, y debe configurar a urn:schemas-microsoft-com:xml-data para poder utilizar la implementación de XML Schema dada por Microsoft. Por ejemplo, podemos tener:

```
<Schema name “EsquemaEjemplo” xmlns=”urn:schemas-microsoft-com:xml-data”>  
Contenido</Schema>
```

Hacemos un pequeño paréntesis para introducir otro de los conceptos más importantes que se manejan en XML, el espacio de nombres:

Espacios de nombres

Los espacios de nombres se emplean para asegurar el significado único entre los nombres de los elementos y los atributos que se están asociando con un determinado vocabulario XML.

Uniendo esta definición con el atributo xmlns, podemos decir que este último es considerado como una orden de procesamiento, porque establece el espacio de nombres de un documento o de un elemento.

Además de asociar el espacio de nombres, es necesario especificar el nombre de los tipos de datos que se van a utilizar. El espacio de nombres del tipo de datos se suele asignar al atributo `xmlns:dt` y configurar a `urn:schemas-microsoft-com:datatypes`. De esta forma podremos usar una serie de tipos de datos. Por lo tanto, en el anterior ejemplo, quedaría:

```
<Schema name="" "EsquemaEjemplo" xmlns="urn:schemas-microsoft-com:xml-data"
xmlns:dt="urn:schemas-microsoft-com:datatypes">Contenido</Schema>
```

El elemento `Schema` puede contener elementos secundarios: `AttributeType`, `ElementType` y `description`. Los dos primeros definen tipos de atributos y elementos respectivamente. El elemento `description` sirve para introducir documentación.

Elemento datatype

Se utiliza para especificar el tipo de datos de un tipo de elemento o atributo. En XML Schema los elementos y atributos se definen por tipos de elementos y atributos, mediante los elementos `ElementType` y `AttributeType`. El elemento `datatype` incluye un atributo, `dt:type`, que se usa para especificar el tipo de elemento o de atributo. Los valores que puede tomar el atributo `dt:type` son: *entity*, *entities*, *enumeration*, *id*, *idref*, *idrefs*, *nmtoken*, *nomtokens*, *notation*, *string*, *bin.base64*, *bin.hex*, *boolean*, *char*, *date*, *dateTime*, *dateTime.tz*, *fixed.14.4*, *float*, *int*, *number*, *time*, *time.tz*, *i1*, *i2*, *i4*, *r4*, *r8*, *ui2*, *ui4*, *uri* y *uuid*.

Elemento ElementType

Es utilizado para definir los tipos de elementos que establece el esquema de los documentos. Junto con el elemento `AttributeType` es el objeto que forma el esqueleto de cualquier XML Schema.

Puede contener elementos secundarios tales como:

- "Datatype": tipo de datos del tipo de elemento.
- "Element": instancia de un elemento secundario del elemento.
- "Group": agrupación de elementos con fines organizativos.
- "AttributeType": define un tipo de atributo.
- "Attribute": identifica un atributo del tipo de elemento.
- "Description": incluye la descripción de un elemento.

Todos los tipos de atributos que se definan dentro de un "ElementType" son locales a ese elemento, es decir, en diferentes elementos pueden existir tipos de atributos con el mismo nombre.

De esta forma, si se quiere definir un tipo de elemento, o un tipo de atributo a nivel global, deberá hacerse justo después del elemento `Schema`.

Ejemplo [17]:

```
<?xml version="1.0"?>
<Schema name="SampleSchema" xmlns="urn:schemas-microsoft-com:xml-data"
xmlns:dt="urn:schemas-microsoft-com:datatypes">
  <ElementType name="PublisherID" model="closed" dt:type="ID"
  content="textOnly">
  </ElementType>
```

...

</Schema>

Dentro de los atributos que se pueden definir dentro de un ElementType, se encuentran:

- Name: Necesario. Indica el nombre del ElementType. Debe ser único para los tipos de elementos en cada ámbito correspondiente.
- Model: Indica si el documento de esquema se asocia a un modelo de contenido abierto o cerrado. Abierto, permite definir elementos adicionales en el tipo de elemento que no se declaran en el esquema, proporcionando mayor extensibilidad. Por defecto, se establece asociación a un modelo abierto (esta es una de las principales diferencias con los DTDs, los cuales sólo admiten modelos cerrados).
- Content: establece el tipo de contenido que hay en el tipo de elemento, y que pueden ser:
 - o "Empty" → sin contenido.
 - o "textOnly" → únicamente texto.
 - o "eltOnly" → contendrá los elementos secundarios indicados.
 - o "Mixed" → contendrá tanto texto como elementos secundarios.1983
- Order: establece el orden y la frecuencia del grupo de elementos secundarios que forman el tipo de elemento.
- Dt_type: Establece el tipo de contenido del tipo de elemento, pudiendo tomar los mismos valores que el elemento "datatype".

De igual manera, se pueden diferenciar dos tipos de elementos dentro de XML Schema;

- Tipos simples ("Simple Type")
- Tipos compuestos ("Complex Type")

"Simple Type"

Los tipos simples son aquellos que no admiten atributos dentro de él, y ni siquiera es posible definir atributos en ellos.

"Complex Type"

Al contrario que los anteriores, los tipos complejos pueden tener dentro de si mismos otros elementos, y además es posible definir atributos en ellos.

Elemento AttributeType

Define un tipo de atributo que se va a utilizar en los elementos definidos a su vez en un Schema [17].

```
<AttributeType name="colors" dt:type="enumeration"
dt:values="red green blue">
```

Los atributos que puede llevar un AttributeType son:

- default: Valor predeterminado del atributo.
- dt:type: Tipod e datos para este tipo de atributo.

- dt:values: Lista de valores posibles cuando se define dt:type como *enumeration* .
- name: nombre del tipo de atributo.
- Required: indicador de si el atributo es obligatorio o no.

2.6 Vinculación XML

Un vínculo en HTML, o hipervínculo, siempre establece un origen y un destino, y ambos siempre cumplen el mismo papel.

Partiendo de la simplicidad con que se definieron estos vínculos, con la aparición de XML surgen nuevas necesidades, entre las que se encontrarían:

- Vínculos bidireccionales, eliminando los recursos fijos como origen y destino.
- Multiplicidad de recursos destino.
- Nuevas opciones de visualización.
- Construcción de nuevos documentos a partir de otros documentos mediante los propios vínculos

A raíz de estas necesidades, surgieron nuevas especificaciones desde la W3C, que dan solución al amplio abanico de nuevos requisitos que han ido apareciendo. Estas especificaciones son principalmente tres: XPath, XPointer y XLink.

XPath

Es un lenguaje no XML que se utiliza para dirigirse a partes de un documento XML. Forma la base del direccionamiento en XPointer y XLink.

XPath trata el documento como un árbol de nodos. A continuación mostramos los diferentes tipos de nodos que podemos encontrar en un árbol XPath [16]:

- Nodos raíz
- Nodos de elementos
- Nodos de texto
- Nodos de atributos
- Nodos de espacios de nombres
- Nodos de instrucciones de procesamiento
- Nodos de comentarios

XPointer

Es un lenguaje no XML, basado en XPath que se usa para dirigirse a las estructuras internas de los documentos XML. Soporta una sintaxis concisa que le permite recorrer la estructura de árbol de un documento XML para hacer referencia a los elementos, atributos y cadenas de texto. Es parte importante de XLink (que veremos a continuación), ya que especifica la sintaxis empleada para crear identificadores de fragmentos, que representan a las direcciones en las construcciones internas de documentos. La W3C

identificó una serie de parámetros de diseño, para ilustrar las directrices utilizadas en la creación de XPointer [38]:

- Los XPointer deben estar dirigidos a documentos XML
- Los XPointer deben poder ser usados de forma clara en Internet
- Los XPointer deben poder ser utilizados en una simple manera en los URI
- El diseño de XPointer debe ser preparado de forma rápida.
- El diseño de XPointer debe ser formal y conciso
- La sintaxis XPointer debe ser compacta y legible
- Los XPointer deben ser optimizados para su utilización
- Debe ser factible implementar XPointer

XLink

XLink es un vocabulario XML que permite la vinculación avanzada en XML. Está diseñado para soportar vínculos sencillos, como en HTML, de forma unidireccional, así como una serie de vínculos extendidos que aportan mayor funcionalidad. Mediante XLink se pueden vincular dos documentos a partir de un tercero, haciendo que el vínculo sea de por sí un recurso viable. XLink consta de una serie de parámetros de diseño, definidos por la W3C (al igual que XPointer) [37]:

- XLink debe poder ser utilizado en Internet de forma fácil
- XLink debe poder ser utilizado por una serie de dominios de utilización de vínculos y por clases de software de aplicación de vinculación
- XLink debe soportar las construcciones de vinculación de HTML 4.0
- El lenguaje de expresión de XLink debe ser XML
- El diseño de XLink debe ser formal, conciso e ilustrativo
- Se debe poder leer y escribir en los XLinks
- Los XLinks pueden residir dentro o fuera de los documentos en los cuales residen los recursos implicados.
- XLink debe representar la estructura abstracta y la importancia de los vínculos
- Debe ser posible implementar XLink, independientemente de los medios empleados para la presentación de los vínculos
- XLink debe ser informado de la existencia de sistemas y estándares de hipermedios (contenidos de texto, imagen video...)

Para finalizar, resaltar que una herramienta importante dentro de Xlink es la de “documento linkbase”, utilizada para facilitar la gestión de enlaces, y que es un documento XML que contiene colecciones de enlaces. Esta herramienta tiene especial importancia en el lenguaje de informes XBRL [15].

2.7 Tecnologías XML

Además de las tecnologías de vinculación, existen otros muchos lenguajes y tecnologías que están basadas en XML [40].

- CML (“Chemical Markup Language”): El lenguaje de marcado de química es una aplicación que fue desarrollada por Peter Murray-Rust. Esta aplicación es considerada la primera aplicación de XML. Este lenguaje trata de manipular información química usando XML/SGML y el lenguaje de programación Java.
- MathML (“Mathematical Markup Language”): Este es un lenguaje que permite incluir fórmulas matemáticas en páginas web sin que exista la necesidad de utilizar algún tipo de conversor. La gramática del lenguaje es muy potente permitiendo definir toda la gramática de la aritmética elemental hasta la gramática de cálculo y ecuaciones diferenciales.
- BPEL (“Business Process Execution Language”): El lenguaje BPEL define una notación para especificar el comportamiento del proceso del negocio basándose en WebServices. Esta notación estándar es llamada “Business Process Execution Language for WebServices” (BPEL4WS). Al ser un modelo basado en WebServices los procesos que se describen en BPEL4WS exportan e importan funcionalidades usando solamente interfaces de servicios web. BPEL4WS proporciona una notación formal de los procesos del negocio y protocolos de interacción de negocios.
- SOAP (“Simple Object Access Protocol”): Este lenguaje define el format de los mensajes a ser intercambiados en un servicio web.
- WDSL (“WebServices Description Language”): Completando al lenguaje SOAP, WDSL genera la autodescripción del servicio.
- XML RPC (“Remote Procedure Call”): Fue una de las primeras aplicaciones en crearse, y su función era la de realizar llamadas a procedimientos remotos en Internet.
- SVG (“Scalable Vector Graphics”): Lenguaje que define gráficos vectoriales.
- SMIL (“Synchronized Media Integration Language”): Destinado a la creación de presentaciones multimedia que combinan audio, imágenes, texto y video.
- XQL (“XML Query Language”): Es el lenguaje de consulta sobre XML
- XSL (“XML Style Language”): Lenguaje Extensible de Hojas de Estilo que indica como debe estar estructurado el contenido, como debe ser diseñado de origen y como debe ser presetrnado en destino.
- CSS (“Cascading Style Sheets”): Hojas de estilo en cascada, que cumple la misma función que XSL, pero es anterior y ya se utilizaba con HTML.
- DOM (“Document Objet Model”), SAX (“Simple API for XML parsing”), que son especificaciones que permiten definir procesadores de lenguaje XML.

Existen multitud más, pero es un tema que va evolucionando constantemente, y a día de hoy, se puede decir que XML es una tecnología que todavía no ha llegado a su límite en lo que a explotación se refiere.

3. INTRODUCCIÓN A XBRL

Antes de adentrarnos en el meta metamodelo de XBRL es necesario realizar una introducción a este lenguaje de informes, conocer el contexto histórico a partir del cual surge, y por tanto, descifrar sus orígenes, su nacimiento y su evolución. A continuación, definiremos XBRL, tanto el lenguaje en sí, como su arquitectura, para finalmente ver la importancia y la implantación que existe en nuestros días de este estándar de reporting financiero.

3.1 Contexto histórico

En la actualidad, vivimos en un mundo que ha experimentado grandes cambios en los últimos tiempos. Muchos de estos cambios tienen su origen en un concepto que está en boca de todos, la globalización. Dicho término no puede entenderse sin comprender que su origen está, sin lugar a dudas, en el sector económico.

La esencia del término se basa, sin dejar de lado otros factores, en la vinculación de los diferentes mercados existentes. Esto es, en la interrelación de las economías, a nivel estatal o de los propios consumidores en su papel de individuos o colectivos privados. La globalización implica una apertura en todos los aspectos. Si bien la idea que primero pasa por nuestras cabezas es la de un comercio a nivel mundial, no se pueden omitir otras interpretaciones del término en cuestión. Globalizar es también ampliar el espectro de actuación de todos y cada uno de los apartados que intervienen en la vida política, social, económica, y como no, tecnológica del universo que hoy conocemos. Sin embargo, es imposible tratar por separado cada uno de esos apartados sin detenerse en las relaciones que hacen que todos avancen en el mismo sentido y hacia un mismo objetivo común.

Sirvan como un ejemplo rápido las diferentes organizaciones que han ido surgiendo (y que a su vez han convertido a sus componentes en simples nodos de una red mayor), como pudieran ser la Unión Europea (UE), la Organización de Naciones Unidas (ONU), la Organización del Tratado del Atlántico Norte (OTAN), el fondo Monetario Internacional (FMI), la Organización Mundial de la Salud (OMS), ...

En lo que respecta al mundo tecnológico, el concepto globalización no se puede entender sin la gran red internacional que ha permitido el intercambio de información, a todos los niveles, entre de todos y cada uno de nosotros, Internet.

Evidentemente, no es posible este intercambio de información entre individuos sin establecer previamente una serie de reglas que deban ser aceptadas y respetadas por todos los miembros de la comunicación. Es en este punto donde surge otro concepto que va ligado estrechamente al de globalización: la estandarización.

Estas reglas o normas, no son otra cosa que definiciones de cómo, donde, cuando comunicarse, y por supuesto, qué hay que comunicar. Si estas ideas las llevamos a los diferentes sectores, podemos ver que existen estándares de comunicación para casi todo, por ejemplo, ¿por qué una persona propietaria de una tarjeta de crédito puede sacar dinero en cualquier cajero del mundo? Sencillamente porque existe un estándar definido para ello (ISO 7810), y al que deben acogerse los miembros que quieran participar de dicho servicio [30].

Si nos centramos en el sector económico (recordemos de nuevo, el origen de todo esto), podemos deducir que existirá un estándar para cada comunicación que se quiera establecer entre dos partes. Y si nos paramos en el sector económico, es inevitable unirlo al sector financiero, y a todo su conglomerado de empresas, públicas o privadas que se dedican a la intermediación financiera. En los días que hoy vivimos, no es difícil ver que entidades bancarias, bolsas de valores, empresas de seguros, etc., tienen un papel primordial en la vida económica de todos nosotros.

Si hemos visto que se tiende al intercambio de información entre todas las partes, el mundo tecnológico cobra aquí una gran importancia. Y es que va a ser este sector el que ponga los medios para llevar a cabo el intercambio anteriormente descrito [20].

Intercambio de información, a nivel mundial, regido por unas normas para su universalización. Si lo centramos en el sector financiero, tenemos el punto de inicio de XBRL.

3.2 Orígenes

Una vez establecidas las directrices sobre las que el mundo se mueve, hemos de evaluar las causas por las cuales surgen los estándares que van a ser objeto de estudio en este trabajo. Estas causas son, ni más ni menos, los escándalos financieros que han protagonizado grandes empresas desde hace años, como por ejemplo ENRON (<http://www.enron.com/>).

ENRON fue fundada en el año 1985 a raíz de la fusión entre Houston Natural Gas e InterNorth, dando lugar a la mayor red de transporte de gas de Estados Unidos. En 1989 comienza a comercializar gas natural como “commodity” (productos destinados al uso comercial). Rápidamente se convierte en el mayor comercializador de gas natural de Estados Unidos y del Reino Unido. En 1999 se produce el lanzamiento de “ENRON Online”, un sistema de transacciones globales en internet que permitía a los clientes de ENRON ver en tiempo real los precios del mercado y realizar transacciones en línea, de forma instantánea. En dos años, esta plataforma llegó a registrar 6000 transacciones diarias por un valor de 2.500 millones de dólares. En 15 años ENRON pasó de ser una pequeña firma de gas en Texas a ser la séptima compañía más grande de Estados Unidos, con 21.000 empleados en más de 40 países, superando los 100 billones de dólares americanos de facturación en el año 2000. Su compleja estructura corporativa, con más de 3.000 sociedades unidas a través de holdings, hace que sea más difícil realizar auditorías mediante métodos convencionales, resultando relativamente “fácil” ocultar y manipular resultados. ENRON se diversificó, poseyendo desde plantas de energía hasta distribuidoras de gas, pasando por compañías de agua entre otros servicios. El éxito de ENRON se basó en que incluso este tipo de servicios y productos, como por ejemplo anchos de banda (empresa Telecom), eran en realidad “commodities”, pudiendo ser comprados, vendidos y almacenados tal como se hace con las acciones y bonos. En el año 2001, poco después de que los directivos de ENRON anunciaran más beneficios, se descubrieron graves irregularidades tanto en las cuentas de ENRON como en las auditorías llevadas a cabo por la empresa Arthur Andersen, la cual comienza a destruir documentación de dichas auditorías. En Octubre de 2001, ENRON declara pérdidas por 638 millones de dólares. Sus acciones se derrumban en bolsa. Se descubren que las pérdidas son aún mayores que las declaradas. Finalmente, ENRON quiebra. Los directivos de la empresa venden todas sus acciones justo antes de la misma. No así sus empleados y jubilados, los cuales habían visto como ENRON les premiaba con opciones de compra (“Stock Options”) durante años. Opciones que ahora carecen de valor. Miles de empleados son despedidos. Arthur

Andersen desaparece, tras 89 años de vida, y como resultado de la manipulación de los datos de las auditorías realizadas a ENRON. En 2006 se celebra el juicio por el “Caso Enron”, durante 15 semanas y con 54 testigos. Kenneth Lay (fundador) y Jeffrey Skilling, los responsables del escándalo, son condenados.

El caso de ENRON no resultó ser un caso aislado, ya que a raíz de este escándalo, se descubrieron otros que afectaban a otras grandes empresas, y que a su vez, eran auditadas de manera irregular. Worldcom, el segundo operador de telecomunicaciones americano y Tyco International, entre otras, se encontraron en la misma situación.

Se llegó a la conclusión de que había que aumentar los mecanismos de supervisión y control financiero.

3.3 Nacimiento y Evolución de XBRL

Para entender qué es XBRL, hemos hablado de globalización y estandarización. Para comprender el porqué de la necesidad de un lenguaje como XBRL, hemos hecho referencia a hechos concretos surgidos desde comienzos del siglo XXI:

- En el año 2001 se descubrieron irregularidades financieras en Enron, compañía eléctrica puntera a nivel mundial, que la obligó a disolverse, a despedir a miles de trabajadores, a cuantiosas pérdidas a sus inversoras, y que además llevó consigo la caída de la auditora Arthur Andersen.
- En 2002 fue Worldcom, uno de los mayores operadores de telecomunicaciones americano, llegó a la bancarrota debido a los fraudes fiscales cometidos.
- En es mismo año, Tyco International era investigada.
- ...y así con numerosas empresas más...

No deja de ser sorprendente que, en la época en la que vivimos, la manipulación de datos de empresas, la ocultación de información, sean acciones que estén a la orden del día.

Si ahora, juntamos los términos anteriormente mencionados, globalización y estandarización, y los unimos con estos escándalos financieros, obtenemos el contexto en el que nace XBRL.

XBRL nace oficialmente en 1998, con la primera especificación XML del W3C recién publicada, de la mano de Charles Hoffman [9], quien trabajaba en una pequeña empresa de Tacoma (USA), con mucha experiencia en sistemas financieros automatizados. Empezó a investigar el uso de XML en la presentación de informes financieros y, apoyado por la AICPA [2] (“American Institute of Certified Public Accountants”, colegio de auditores públicos de USA), se aprueba el uso de XML para este fin.

En 1999, y financiado por la propia AICPA, surge XFRML [5] (“eXtensible Financial Reporting Markup Language”), con miembros tales como Arthur Andersen, Ernst & Young, Microsoft Corporation...entre otros. XFRML se utilizó en primera instancia para presentar los estados contables de 10 compañías.

En el año 2000 XFRML cambia su denominación por XBRL (“eXtensible Business Reporting Language”), ampliando su espectro de actuación no sólo a información financiera. En ese mismo año se creó la primera taxonomía, dirigida a representar los informes financieros de las empresas comerciales e industriales (aproximadamente el 80% de las empresas que cotizan en USA).

A partir de esta primera taxonomía, XBRL comienza su expansión, apoyada en tres pilares:

- 1) Alineamiento con los estándares técnicos del W3C.
- 2) Reconocimiento y adopción por las instituciones que pueden darle rango legal (a raíz de Enron).
- 3) La internacionalización.

Las principales organizaciones que han participado y colaborado en el desarrollo (y que forman el anteriormente mencionado, consorcio AICPA) de XBRL se encuentran entre las principales potencias de la industria informática, la contabilidad y la consultoría, además de contar con el apoyo de instituciones como el IASB (“International Accounting Standards Board”), el IMA (“Institute of Management Accountants”), el CICA (“Canadian Institute of Chartered Accountants”) o el ICAEW (Institute of Chartered Accountants in England and Wales)

Profundizando un poco más en el propio lenguaje XBRL, hemos visto que es necesario definir qué conceptos queremos reflejar. Por ejemplo, podrían definirse los totales de gastos e ingresos de cualquier entidad bancaria, el importe total de transferencias realizadas, o cualquier definición de dato que sea necesario reflejar. Esto es lo que llamaríamos metadatos, datos que definen datos. Los hechos sería la información concreta. Para todos estos términos, que podrían coexistir en diferentes entornos, y que evidentemente podrían llamarse de distinta manera, conteniendo exactamente lo mismo, se intenta definir un lenguaje universal. Y para ello, hay que recurrir a referencias legales, ya que de otra manera sería imposible unificar criterios. Por lo tanto, en la definición de un concepto, hay que tener en cuenta que se entiende por concepto en términos legales, que propiedades tiene, y que relaciones con otros términos pueden darse.

Por otro lado, para describir el dato en sí, hace falta saber a qué concepto hace referencia, en qué contexto se está utilizando, y por último, qué valor tiene. Lo primero no es otra cosa que la etiqueta que se anteponga al dato, el identificador propiamente dicho. El contexto incluye un periodo de tiempo, el propietario y la lógica de negocio del propio contexto. Por último tenemos el valor, qué debe ser un dato cuantitativo, es decir, el hecho.

Los metadatos están definidos en las taxonomías, junto con sus adjetivos, como son las etiquetas, referencias, etc, y sus restricciones.

Los esquemas de las taxonomías tienen su base en un XML Schema, es decir, en los metadatos de XML. Igualmente, se representarán los conceptos mediante elementos, con su espacio de nombres correspondiente. Así mismo, tendremos atributos y cualquier otro objeto que pueda ser utilizado en XML.

La complejidad del mundo empresarial obligó a hacer de XBRL un lenguaje que pudiera verse ampliado continuamente. Los diferentes tipos de usuarios que pueden hacer uso del lenguaje provocaron que las taxonomías fueran extendiéndose, relacionándose entre ellas, heredando conceptos unas de otras, o completando la información heredada, para adaptarla a los requisitos de cada país, sector o empresa.

3.4 ¿Qué es XBRL?

La definición “oficial” dada para XBRL (“eXtensible Business Reporting Language”) explica que es un lenguaje para la comunicación electrónica de datos financieros y de negocio. En un marco de actuación internacional, tiene como fin último el estandarizar la presentación de la información empresarial [3].

Dicho de otra forma XBRL es un estándar abierto que soporta el modelado de información y define el significado semántico que requieren los informes financieros.

Como hemos visto, está basado en XML, utilizando su sintaxis y tecnologías relacionadas con XML, como pueden ser XML Schema, XLink, XPath, o los espacios de nombres. Adicionalmente, al hablar de lenguaje estándar, está el trabajo de estandarización de los conceptos empresariales que se quieran tratar.

Por tanto, se puede decir que XBRL es un lenguaje específico, derivado de XML, cuyo objetivo es satisfacer las exigencias propias de la elaboración y transmisión de informes y formularios relacionados con información contable y administrativa, en su más amplia acepción [35].

La idea básica con la que trabaja XBRL es la base sobre la que lo hace XML. Un dato, identificado por una etiqueta. Evidentemente, no es tan sencillo, existen relaciones, lenguajes, reglas. XBRL trata todos ellos mediante objetos definidos dentro del concepto global.

Como hemos visto, la estandarización ha llegado a todos los niveles. Si a nivel tecnológico, y de intercambio de ficheros, hemos explicado qué es XML, si concretamos y nos centramos en el sector financiero, tenemos que hablar de XBRL.

En la actualidad, la versión vigente de XBRL es la 2.1, desde 2003, ya que aunque han surgido nuevas extensiones y nuevas funcionalidades, no se ha modificado el núcleo central del estándar [34].

Pero, ¿por qué es necesario XBRL? Sencillamente porque facilita la gestión de la información. No es lo mismo presentar 10 documentos de resultados anuales, en 10 formatos distintos, que presentarlos todos de acuerdo a una semántica, a unas reglas comunes, y lo que es más importante, con validación sintáctica y semántica en origen. Su control y visualización es mucho más óptima. Si esto lo extrapolamos a miles de empresas repartidas por todo el mundo, a organismos centrales que deben recoger dichos resultados, o a los diferentes áreas de una misma empresa, la mejora es realmente importante.

XBRL aporta otras grandes mejoras a la hora de gestionar dichos informes financieros, reduce el tiempo de la mayoría de los procesos involucrados, ampliando la calidad del trabajo, y lo que es más importante y consecuencia de estas dos mejoras anteriores: reduce el coste del proceso global [34]. En la figura 2 [32] podemos observar esta reducción.

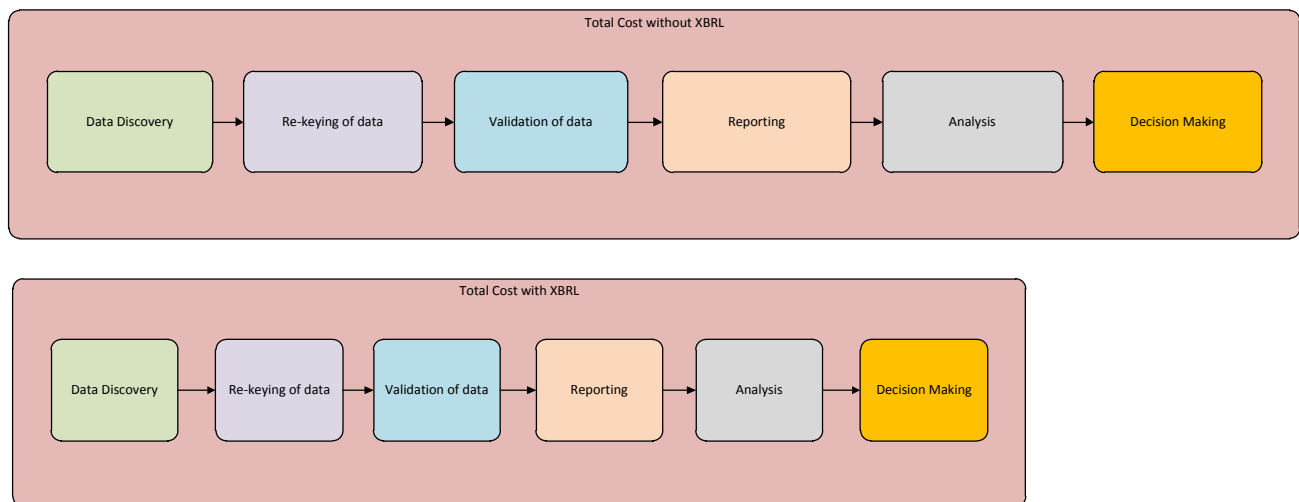


Figura 2. Optimización XBRL

Se debe señalar que la información semántica expresada mediante el lenguaje de informes XBRL es independiente de cualquier aplicación de software. El creador de la información puede proporcionar las reglas de negocio correspondientes utilizadas para expresar la información al destinatario o consumidor de dicha información.

Otra de las características de XBRL, y una de las más importantes, es que es un lenguaje validado en origen. Es decir, el destinatario del informe conoce a priori que el documento está correctamente construido, lo que reduce el coste de comprobación.

En lo que a la organización de XBRL se refiere, existe un organismo a nivel internacional, denominado “XBRL International” [34], sin ánimo de lucro, donde forman parte más de 450 compañías a nivel mundial. La organización Internacional de XBRL es el foro donde se aprueban las taxonomías XBRL, y donde se establecen las pautas a seguir para la evolución del lenguaje.

XBRL Internacional está formado por múltiples organizaciones de XBRL nacionales, o jurisdicciones XBRL, en las cuales hay inscritas más de 250 organizaciones. La jurisdicción XBRL española se fundó en 2004, con el nombre de Asociación XBRL España para el desarrollo de estándares tecnológicos [3], y formaron parte en un principio organismos tales como el Banco de España, Comisión del Mercado de Valores (CNMV), Colegio de Registradores, consultoras, Asociación Española de Cajas de Ahorro (AECA) y empresas de evaluación de riesgos crediticios, de servicios, y de tecnología. Actualmente, ya hay veinticuatro socios, Bancos como BBVA y Popular, compañías de software, hardware, comunicaciones, etc..., y se espera que sigan aumentando en el tiempo.

3.5 Lenguaje XBRL

Hemos visto que XBRL da solución a las necesidades de reporte de toda la información empresarial. En estos informes quedan reflejadas dos partes, los conceptos y los hechos que se ajustan a esos conceptos. La primera parte, recibe el nombre de Taxonomía, la segunda, Documento Instancia, y es uno de los pilares sobre los que se asienta este lenguaje.

Una taxonomía es una ciencia que trata de los principios, métodos y fines de la clasificación [24]. En XBRL, una taxonomía es el conjunto de documentos donde se definen los conceptos que posteriormente se verán representados en los informes, como las relaciones entre ellos. En otras palabras, este conjunto de conceptos se verán presentados de una forma estructurada junto a los datos. El concepto de taxonomía es equivalente al de “XBRL Schema”, y por tanto, es idéntico al de XML Schema, pero aplicado a los conceptos financieros.

Una taxonomía esta formada por (Figura 3):

- Uno o varios esquemas XML denominados esquemas taxonómicos.
- Descripciones de las relaciones entre los elementos de los esquemas taxonómicos, y que reciben el nombre de “documentos linkbase”. Estas relaciones pueden ser entre conceptos o entre conceptos y recursos externos.

Entre estos documentos linkbase, que definen los roles de los conceptos, existen cinco tipos [25]:

- “Label”, o rol de etiquetas, que permite asignar una etiqueta a un determinado concepto, incluyendo el idioma, permitiendo la internacionalización de XBRL. En otras palabras, define los textos que identifican los hechos, por ejemplo:
 - o Total Activo = 1.000.000
- “Reference”, o rol de referencias, que asocian a cada concepto una referencia a un recurso que aporta documentación acerca de su significado. Esa referencia debe asociar un texto legal, aportando información adicional del concepto, pero contiene sólo la referencia al texto legal no el propio texto.
- “Presentation”, usado para especificar el orden en el que se presentan los conceptos. A diferencia de XML, donde existe una gran diferencia entre información y presentación, en XBRL esta última acaba desempeñando un papel importante. A veces, resulta imprescindible establecer un orden, para hacerlo entendible para todos los potenciales usuarios, y por otra amoldarse a los requisitos legales o de usuario. Este rol permite a las herramientas de creación o visualización presentar de manera más “amigable” el contenido, aunque no es un rol de renderización exactamente. Este rol presenta una estructura jerárquica, y los elementos hijos se relacionan con los padres por medio de Xlink.
- “Calculation”, o rol de cálculo usados para proporcionar comprobaciones y métodos a utilizar para la realización de cálculos (*obtención de totales, por ejemplo*). Los elementos padres serán el resultado de operaciones aritméticas realizadas sobre los elementos hijo. Este rol se utiliza sólo para validar conceptos.
- “Definition”, o rol de definición, que establece relaciones entre los elementos de una taxonomía, que permitan explicar o documentar sus relaciones. Además, permite añadir ciertas reglas que son utilizadas en la validación de los documentos XBRL.

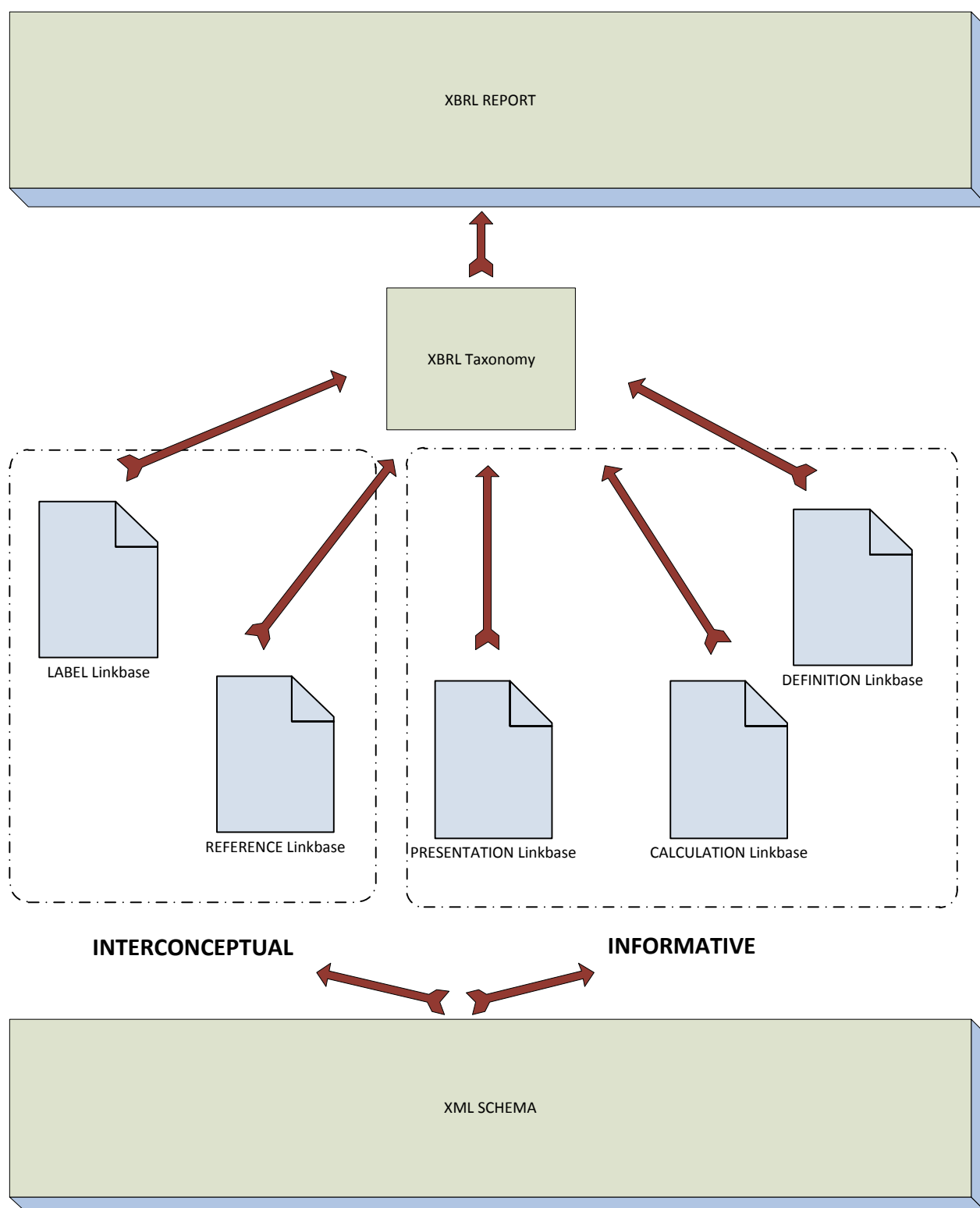


Figura 3. Taxonomía y roles

Todas las taxonomías XBRL están publicadas en la página oficial de XBRL International, y será este organismo quien la califique, bien como “Taxonomía reconocida” (que cumplen con la especificación XBRL

vigente), bien como “Taxonomía aprobada” (además de cumplir con la especificación cumple las directrices oficiales marcadas por XBRL [3].

Cada jurisdicción nacional puede contener sus propios conceptos contables, por lo que es normal que existan taxonomías específicas para cada una de ellas. Además, cada empresa u organismo privado puede tener necesidades muy concretas, por lo que también podrían tener sus propias taxonomías. Para estos últimos casos se ha implantado la taxonomía llamada Libro Mayor (“General Ledger”) [18].

Una vez definido el término de Taxonomía, pasemos a analizar los Documentos Instancia. Son documentos XML, con una colección de hechos, que son elementos que se ajustan a una taxonomía. Estos hechos pueden expresarse de dos formas [15]:

- Ítems: Contenido simple en forma de una sola medida o de un contenido textual.
- Tuples: Contenidos compuestos, en los cuales el valor está formado por otros hechos, simples o compuestos.

Dentro de los primeros, encontramos los “Primary Items”, que son aquellos conceptos que definen el tipo de hecho, y necesita de dimensiones para completar su significado semántico.

Dentro de XBRL, al igual que ocurre en XML, existen Espacios de Nombres específicos, que permiten obtener las etiquetas necesarias para cada informe, y que son [15]:

- xml: <http://www.w3.org/XML/1998/namespace>
- xsi: <http://www.w3.org/2001/XMLSchema-instance>
- xlink: <http://www.w3.org/1999/xlink>, aquí se definen atributos procedentes de la especificación XLink de XML.
 - o xl: <http://www.xbrl.org/2003/XLink>, donde se definen los tipos usados en XBRL
 - o link: <http://www.xbrl.org/2003/linkbase>, en el cual se definen los distintos tipos de enlaces involucrados en XBRL.
 - o xbrli: <http://www.xbrl.org/2003/instance>, donde se especifican los documentos instancia.

Los tres últimos hacen referencia directamente de W3C, mientras que los primeros de XBRL.com. Estos espacios de nombres trabajan de forma jerárquica, similar al concepto de herencia en programación orientada a objetos.

3.6 Arquitectura XBRL

La arquitectura de XBRL se fundamenta en XML, esto es, los documentos XBRL tendrán una estructura de documentos XML. Otra parte importante es la especificación XBRL vigente, ya que deberá ajustarse a sus requisitos y normas. Por último, el informe XBRL [Figura 4] estará formado por diferentes niveles [14]:

- Contenido del informe, formado por un conjunto de hechos contables que se ajustan a la normativa vigente.
- Definición de conceptos y de las relaciones externas e internas de los propios conceptos (taxonomías)

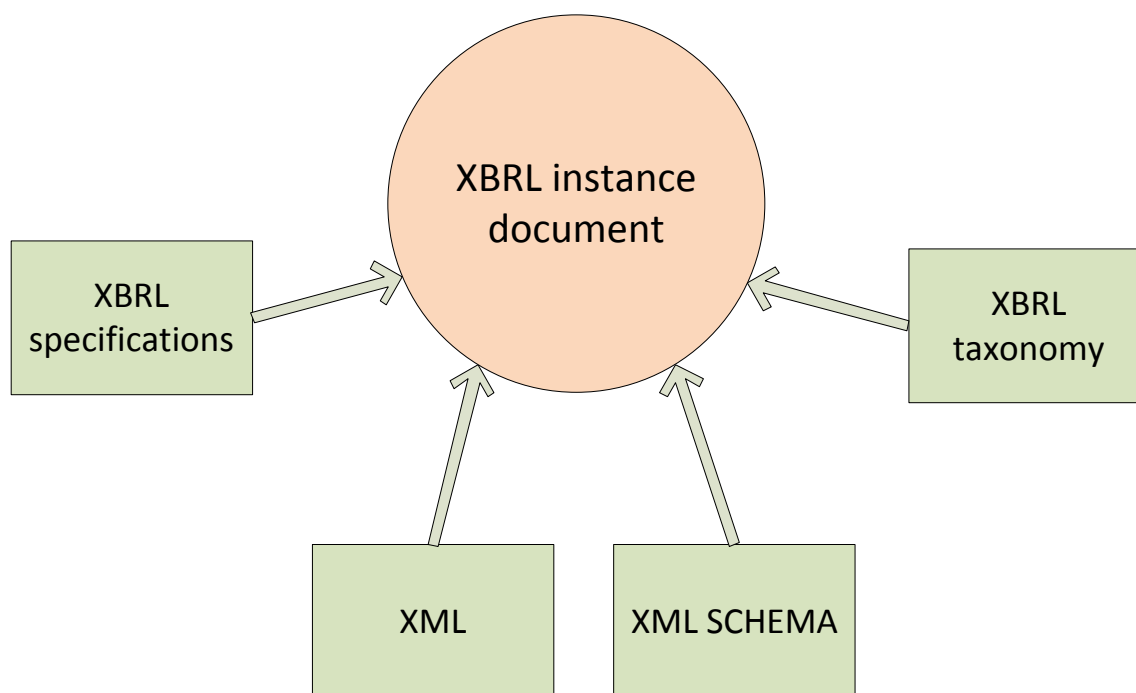


Figura 4. Bases de un documento Instancia

Igualmente, es necesaria una arquitectura funcional para cualquier parte involucrada en la gestión y procesamiento de datos en formato XBRL, que deberá ser capaz de gestionar la creación, distribución y uso de diferentes taxonomías, así como la creación, publicación, recepción, validación e interpretación de instancias. Así mismo, deberá poseer un repositorio para almacenamiento de los datos y para búsquedas.

Además, el trabajo con XBRL implica otros componentes adicionales como mecanismos de seguridad, herramientas para el propio desarrollo de taxonomías, herramientas de tratamiento de informes XBRL, funciones de análisis de información XBRL o procesos batch de instancias XBRL.

Hay que olvidarse que otra parte de la arquitectura de XBRL son los conceptos contables o financieros manejados en cada sector, ya que los conceptos son conceptos independientemente del dominio de ellos.

Adentrándonos un poco más en esta arquitectura, podemos diferenciar cuatro capas: la capa conceptual, donde se definen los conceptos, tanto su sintaxis como su semántica, la capa de relaciones, donde se detallan todas las relaciones entre conceptos, la capa DTS (Discovery Taxonomy Set), que auna taxonomías y documentos linkbases para interpretar un documento instancia. DTS es el conjunto de XBRL Schemas o XML Schemas y sus ficheros XML asociados o roles que constituyen una taxonomía y validan un documento instancia XBRL.

La Capa Conceptual

En esta primera capa es donde vemos los conceptos, que son las declaraciones de los elementos en una taxonomía. Por otro lado, está la definición de un hecho dentro de un informe. Los conceptos pueden estar relacionados con otros conceptos o recursos, a través de enlaces XLink (role). De la misma forma, para cada concepto definido, debe existir una etiqueta que lo identifique (como en XML), en cualquier idioma. Además, puede existir información complementaria al concepto que ayude a su comprensión. En las

taxonomías debe definirse únivocamente cada concepto, evitando cualquier ambigüedad, por lo que cada etiqueta debe ser concreta [15].

Los documentos instancias son parte de esta capa, ya que son hechos concretos de los conceptos, en un determinado contexto. Las etiquetas y las referencias que hemos comentado, también. Son roles que poseen los conceptos, que los dan forma y los enlazan con recursos para completarlos (Figura 5, adaptada de la referencia [15]).

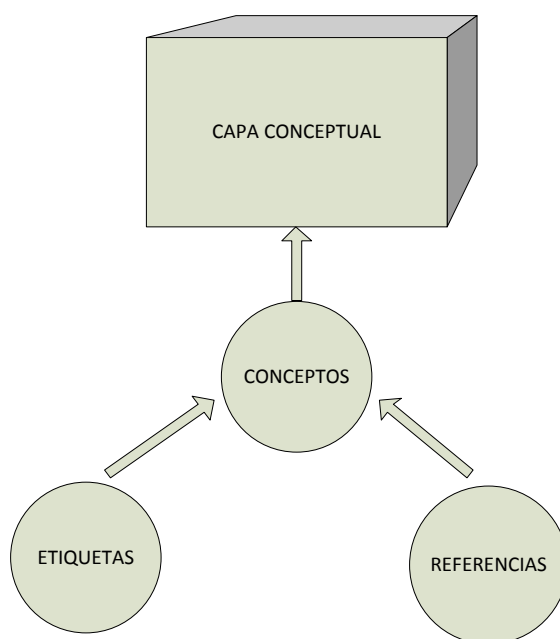


Figura 5. Roles de la capa conceptual

La Capa de Relaciones

Las relaciones en XBRL juegan un papel fundamental, ya que son ellas las que uniendo tanto hechos como conceptos, consiguen dar un significado único. Los arcos son los elementos que enlazan un origen con un destino, permitiendo dar mayor complejidad y profundidad a dichos enlaces. De esta capa forman parte las propias definiciones de los conceptos, el orden en que se presentan y las fórmulas utilizadas para calcular aquellos campos que corresponen (Figura 6, adaptada de la referencia [15]). Son relaciones interconceptuales, esto es, como hemos dicho, entre los propios conceptos, y entre los roles entre sí [15].

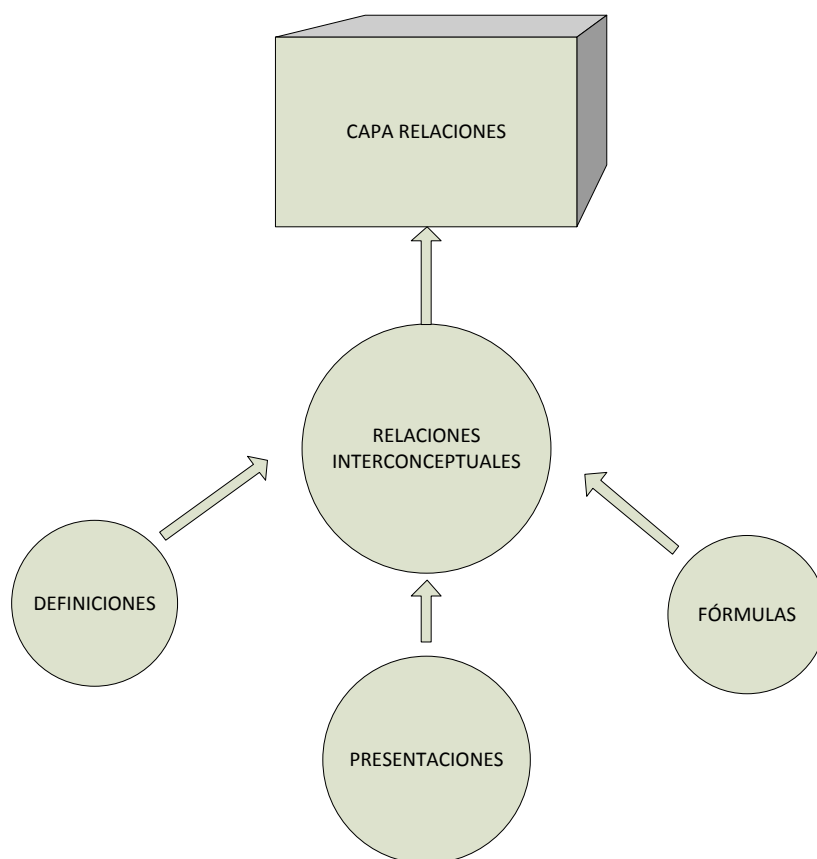


Figura 6. Roles de la capa de relaciones

La Capa “Discoverable Taxonomy Set” (DTS)

La extensión del lenguaje XBRL ha provocado que una taxonomía no sea suficiente para interpretar un informe XBRL, si no que se necesiten varias que se relacionen entre si, y que estén correctamente construidas y jerarquizadas. Esto es debido a que las taxonomías se heredan, de tal forma que puede haber una definición de conceptos generales, como la taxonomía del International Financial Reporting Standards (IFRS), luego, por ejemplo, el Supervisor Europeo oportuno, adapta a su normativa esa definición de conceptos con otra taxonomía, y a su vez cada país adapta dicha definición de conceptos a su normativa. Para la creación o derivación de todas estas taxonomías deben emplearse metodologías que mantengan el significado unívoco de los conceptos durante la interpretación de los documentos instancia. Un DTS no es otra cosa que un conjunto de fichero XML Schema asociados a un fichero XBRL (un documento instancia) [Figura 7]. El informe XBRL tendrá lugar y se validará por y contra ese conjunto de taxonomías.

Un ejemplo práctico puede ser una taxonomía generada o validada por IFRS (www.ifrs.org), que posteriormente es adaptada en España por la CNMV. El informe correspondiente estará relacionado con ambas taxonomías. Además, cada una de las taxonomías tendrá sus propios roles.

Todo esto está detallado por XBRL International [34], permitiendo que cada DTS pueda ser utilizado por varias aplicaciones [15].

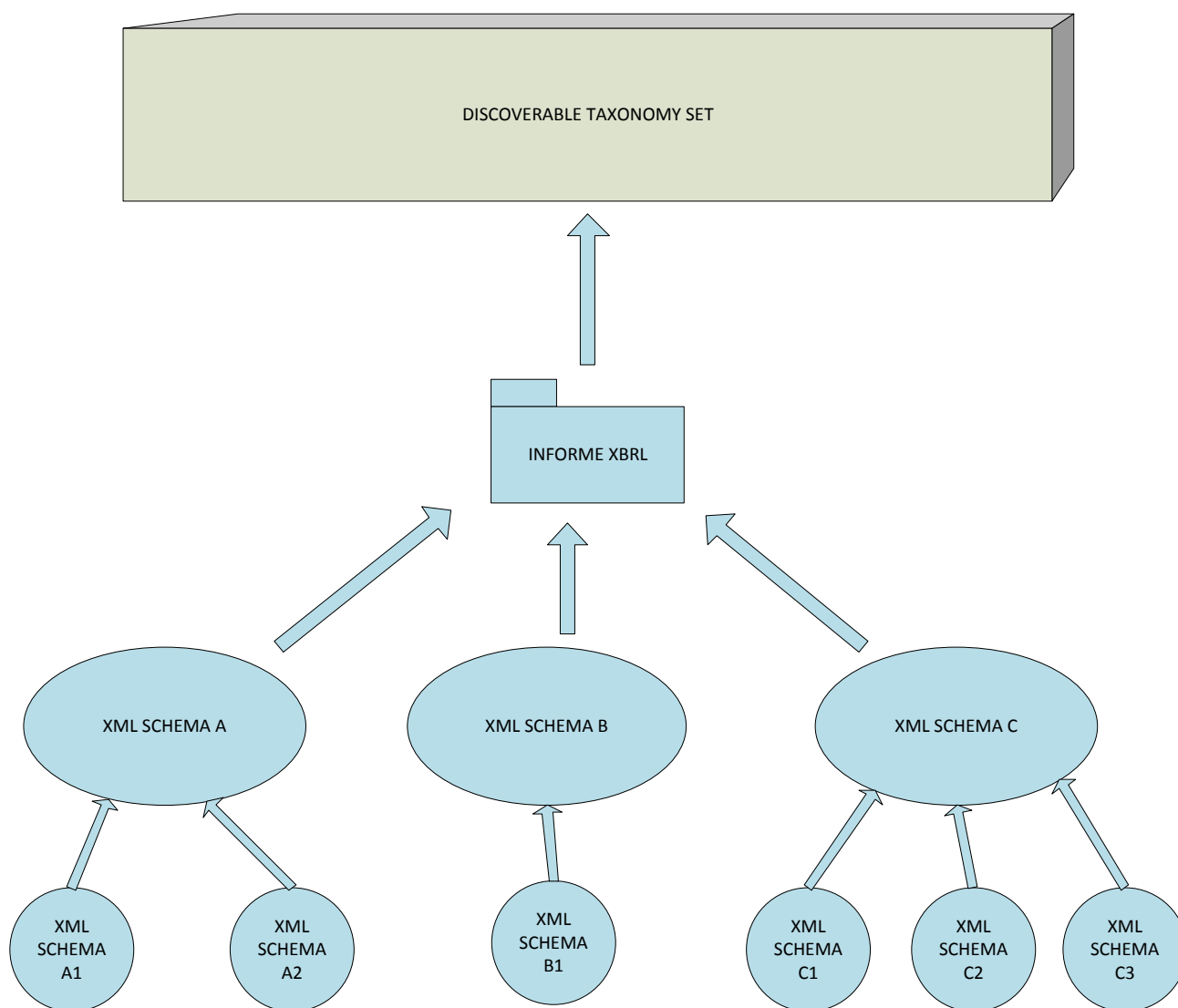


Figura 7. Ejemplo de estructura de DTS

La Capa de Extensiones

Debido precisamente a la extensión de XBRL, las taxonomías han ido aumentando en complejidad casi proporcionalmente. La extensión de taxonomías supone seguir una metodología adecuada, ya que de otra forma, y dada la dificultad del proceso, no habría manera de poder trabajar con documentos que contienen tantos elementos que en ocasiones podrían resultar inmanejables.

El concepto de “extensión” no se corresponde sólo con el de ampliación, si no que también se incluyen actualizaciones y modificaciones. Todo ello provoca que cuando un concepto se extiende, sus relaciones y enlaces deben hacerlo de forma paralela [15]. En otras palabras, los archivos XML Schema son modificados, o nuevos XML Schema son añadidos a una taxonomía [Figura 8].

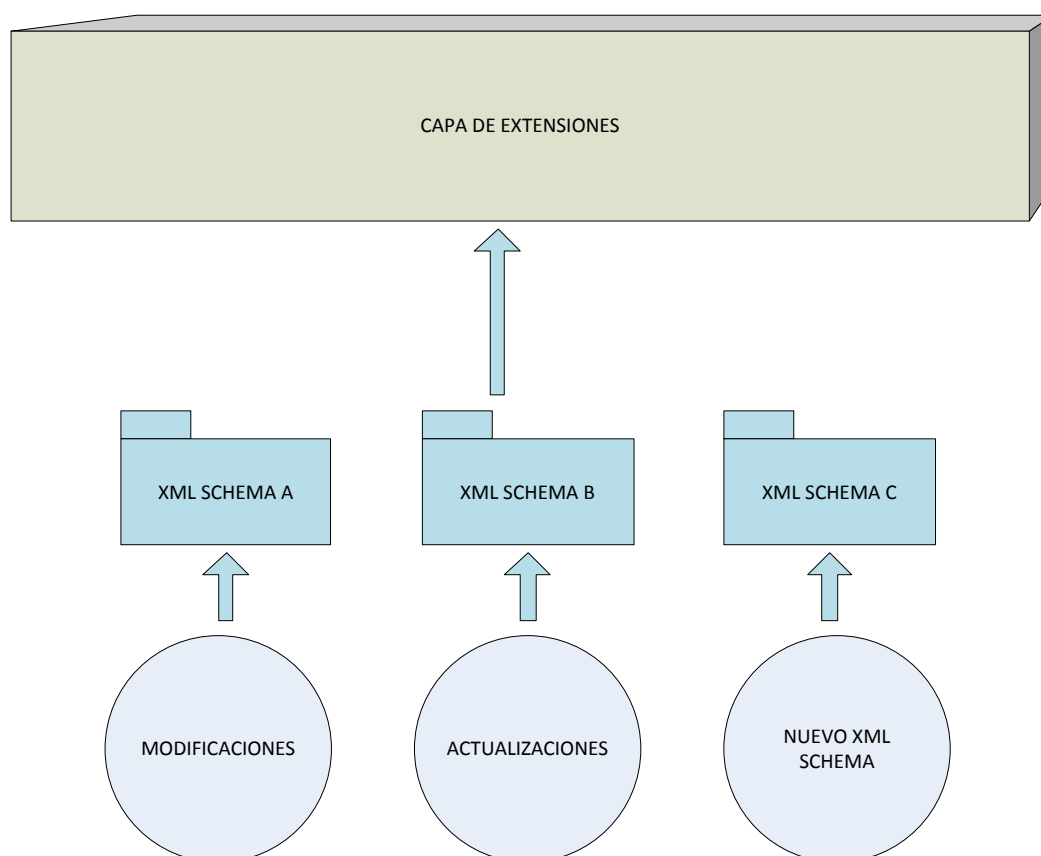


Figura 8. Capa de extensiones

3.7 Proyectos XBRL en el mundo

El uso de XBRL se está extendiendo rápidamente por todo el mundo. Actualmente, existen tres focos principales, centrados en 3 continentes: América, Asia y Europa.

En Asia, por ejemplo, XBRL se emplea para los mercados de capitales. Las bolsas de valores de China (http://www.sse.com.cn/sseportal/en_us/ps/about/news.shtml), Japón, Singapur y Corea del Sur se mueven mediante XBRL. La Agencia de Servicios Financieros de Japón requiere que todas las compañías públicas envíen sus informes financieros en formato XBRL a partir de este año. Empresas privadas como Futjisu han comenzado a beneficiarse del uso interno de XBRL para sus propias aplicaciones (XWand). China fue, en 2004, el primer país asiático en adoptar formalmente informes XBRL para sus mercados bursátiles. A su vez, sigue siendo la principal fuerza para asentar este estándar en otras áreas como perfiles de riesgo o minería de datos.

En el continente asiático no existe un organismo central capaz de controlar la interoperabilidad con XBRL, por lo que la extensibilidad del formato a veces se vuelve en su contra, provocando un aislamiento a la hora de generar taxonomías. XBRL International tiene la misión de reconocer las taxonomías que se van creando en las diferentes jurisdicciones nacionales y regionales, y de velar por la calidad de las mismas que se encuentran en desarrollo. Sin embargo en Asia no existe un control tan exhaustivo hoy en día.

Mientras, en Europa, la adopción de XBRL es donde ha calado más fuerte, y donde mejor se está implementando. Desde hace 5 años, ha ido en aumento el número de áreas que han comenzado a utilizar XBRL, pertenezcan al sector público o al privado.

Diferentes grupos de desarrollo están avanzando en diferentes países. Veamos algunos ejemplos [17]:

- En Bélgica es obligatoria presentar al Banco Central las cuentas en formato XBRL.
- En Francia el Banco de Francia utiliza XBRL para normalizar el sector de la banca.
- Alemania permite presentar los informes en XBRL a las pequeñas y medianas empresas, a la vez que se extiende su uso en grandes compañías del ámbito privado.
- Italia está comenzando a implantar XBRL.
- Holanda, mediante el uso de XBRL, ha proyectado reducir en un 25% el coste del proceso de auditar la información presentada por las empresas.
- En España, el sector de la banca ya está utilizando el estándar para la presentación de informes financieros al Banco de España. Desde 2005 se puede presentar documentación ante la CNMV en este formato. Se han generado numerosas taxonomías como las referentes al Plan General Contable 1990 (ICAC) [22], Datos Generales de Identificación (Registro Mercantil) [4], Sociedades y Servicios de Tasación (Banco de España), Prevención del Blanqueo de Capitales (SEPBLAC), Instrucciones de contabilidad local (Sector Público), Estadísticas estructurales (INE) y Responsabilidad social corporativa (AECA), las cuales están ya terminadas o en diversas fases de su desarrollo. Se han publicado, desde el grupo de trabajo XBRL del Comité de Supervisores Bancarios (CESB), las taxonomías europeas Common Reporting (COREP) de información sobre solvencia de entidades financieras (Basilea II) y Financial Reporting (FINREP) de estados financieros de entidades de crédito para organismos supervisores (IFRS/NIIC). Estas taxonomías se están utilizando en Alemania, Bélgica, Bulgaria, España, Estonia, Francia, Finlandia, Grecia, Hungría, Islandia, Italia, Lituania, Luxemburgo, Países Bajos, Polonia, y Suecia. También desde España se está impulsando la taxonomía CBSO-RF (European Committee of Central Balance – Reporting Financial) de Centrales de Balances Europeas, la taxonomía FLIPA de Seguros de Crédito para la Asociación Internacional del Seguro de Crédito, y la investigación académica en XBRL para la propuesta de “Enhanced Business Reporting”.
- Reino Unido ha establecido 2011 como año en el que el uso de XBRL es obligatorio para diferentes presentaciones.
- Austria, Chipre, Finlandia...son países donde el lenguaje XBRL se va extendiendo a un cierto ritmo y donde se está implantando progresivamente.

En América, y para no perder posiciones respecto a Europa y Asia, se han hecho grandes inversiones para adaptar a XBRL multitud de plataformas, pasando de ser un programa experimental y de investigación a una realidad empresarial. La Reserva Federal (FED) o el Fondo de Garantía de Depósitos (FDIC) ya están aplicando XBRL en sus comunicaciones con las instituciones financieras. Por otro lado, la Comisión del Mercado de Valores (SEC) ha realizado un gran desembolso para actualizar sus bases de datos a través de XBRL (proyecto EDGAR). Miembros del Consejo Federal de Control de Instituciones Financieras (“Federal Financial Institutions Examination Council”, FFIEC), la Corporación Federal de Seguros de Depósitos (“Federal Deposit Insurance Corporation”, FDIC), el Sistema de la Reserva Federal (“Federal Reserve System”, FRS) y la Oficina del Controlador de la Divisa (“Office of the Comptroller of the Currency”, OCC), disponen de soluciones XBRL para su proceso trimestral de informes financieros bancarios.

4. META-METAMODELO XBRL

A continuación se definirá el meta-metamodelo de datos que utiliza el lenguaje XBRL. Lo llamamos meta-metamodelo porque estamos hablando del nivel más bajo, es decir, estamos definiendo todos los elementos, tipos, atributos y grupos de atributos que forman el lenguaje XBRL. A partir de ellos se definirán los objetos propios para cada taxonomía, y posteriormente tendremos un modelo para un determinado informe. Para ello, es necesario describir brevemente los principales objetos XML que son utilizados, con lo que tendremos una base para entender los distintos objetos XBRL [23]. Posteriormente entraremos en el detalle de cada elemento, tipo, atributo, grupo de atributo y sus relaciones, cardinalidades y subestructuras que lo forman, para de esta forma obtener un meta-metamodelo que nos permita establecer las bases de futuros estudios sobre XBRL.

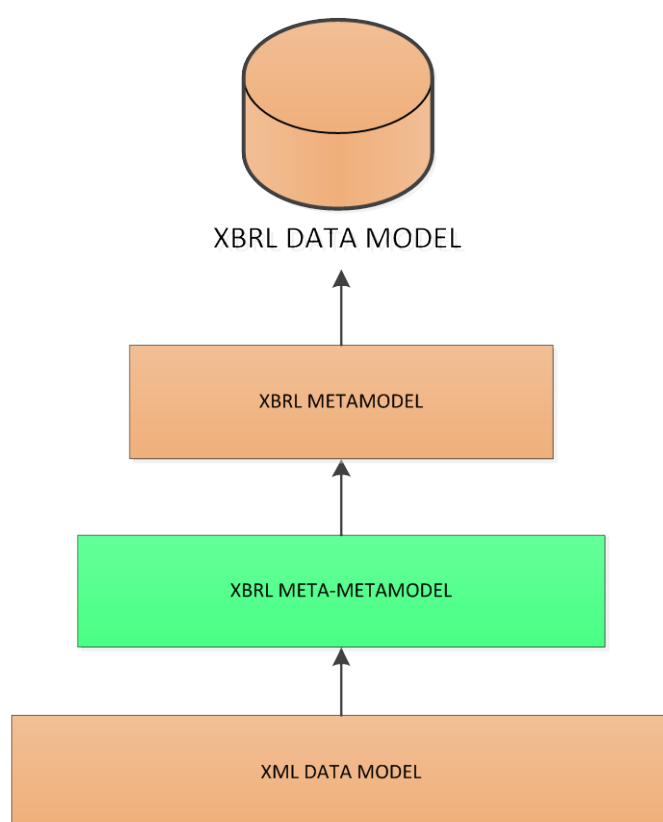


Figura 9. Niveles de modelado XBRL I

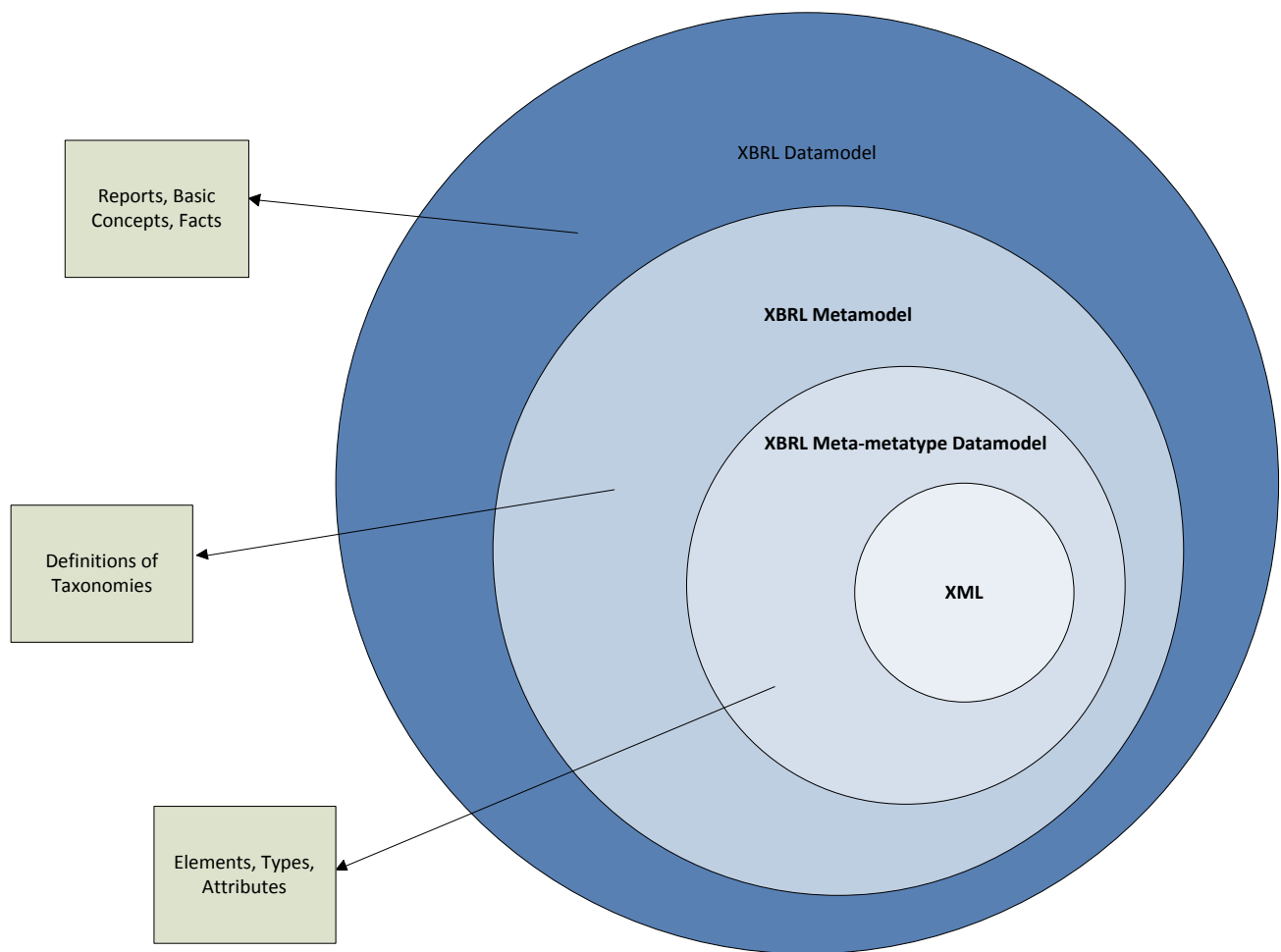


Figura 10. Niveles de modelado XBRL II

4.1 Objetos XML

Primeramente, vamos a realizar una introducción de aquellos objetos XML que son necesarios para la posterior comprensión del modelo de meta-metadatos de XBRL, esto implica que no se van a explicar todos los objetos XML, si no solo los que se utilizan en este estudio del modelo de meta-metadatos XBRL.

Objeto “simpleType”

Un tipo simple es aquel que no podrá contener elementos ni atributos. Pueden ser:

- Tipos predefinidos de XML: string, boolean, integer...
- Listas: listas de datos separados por espacios
- Uniones: Tipo de datos derivado de la unión de tipos predefinidos.

Ejemplo [7]:

```

<xsd:simpleType name="formatoclasificacion">
  <xsd:restriction base="xsd:string">
    <xsd:enumeration value="Trabajo"/>
    <xsd:enumeration value="Familia"/>
    <xsd:enumeration value="Personal"/>
  </xsd:restriction>
</xsd:simpleType>

```

Objeto “complexType”

Un tipo complejo es aquel que permite definir elementos con estructura complejo, es decir, que pueden tener elementos hijos y atributos.

Ejemplo [33]:

```

<xs:complexType name="length1">
  <xs:simpleContent>
    <xsd:extension base="xs:nonNegativeInteger">
      <xsd:attribute name="unit" type="xs:NMTOKEN"/>
    </xsd:extension>
  </xs:simpleContent>
</xs:complexType>

<xs:element name="width" type="length1"/>
<width unit="cm">25</width>

```

Objeto “element”

Los elementos en XML son estructuras con contenido, basado en un tipo definido en la DTD del documento. Puede contener otros elementos, atributos, texto o todos ellos.

Ejemplo [12]:

```

<xsd:complexType name="PurchaseOrderType">
  <xsd:sequence>
    <xsd:element name="shipTo" type="USAddress"/>
    <xsd:element name="billTo" type="USAddress"/>
    <xsd:element ref="comment" minOccurs="0"/>
    <xsd:element name="items" type="Items"/>
  </xsd:sequence>
  <xsd:attribute name="orderDate" type="xsd:date"/>
</xsd:complexType>

```

Objeto “attribute”

Un atributo en XML es un objeto que aporta información adicional sobre un elemento. Un atributo puede constar a su vez de tres atributos:

- type: Especifica el tipo del atributo. Debe ser el nombre de un tipo de atributo ya declarado
- default: El valor predeterminado del atributo
- required: Indicador de opcionalidad u obligatoriedad del atributo

Ejemplo [31]:

```
<person sex="female">
  <firstname>Anna</firstname>
  <lastname>Smith</lastname>
</person>
```

Objeto “simpleContent”

Objeto que indica restricciones o extensiones sobre un “complexType” o un “simpleType” de tipo texto, y que a su vez no podrá contener elementos.

Ejemplo [31]:

```
<xs:element name="shoesize">
  <xs:complexType>
    <xs:simpleContent>
      <xs:extension base="xs:integer">
        <xs:attribute name="country" type="xs:string" />
      </xs:extension>
    </xs:simpleContent>
  </xs:complexType>
</xs:element>
```

Objeto “complexContent”

Objeto que indica restricciones o extensiones sobre un “complexType” de contenido mixto o elementos.

Ejemplo [31]:

```
<xs:element name="employee" type="fullpersoninfo"/>

<xs:complexType name="personinfo">
  <xs:sequence>
    <xs:element name="firstname" type="xs:string"/>
    <xs:element name="lastname" type="xs:string"/>
  </xs:sequence>
</xs:complexType>
```

```

<xs:complexType name="fullpersoninfo">
  <xs:complexContent>
    <xs:extension base="personinfo">
      <xs:sequence>
        <xs:element name="address" type="xs:string"/>
        <xs:element name="city" type="xs:string"/>
        <xs:element name="country" type="xs:string"/>
      </xs:sequence>
    </xs:extension>
  </xs:complexContent>
</xs:complexType>

```

Objeto “attributeGroup”

A través de este objeto es posible pre-definir un grupo de atributos para ser empleados al declarar un elemento “complexType” o “simpleType”.

Ejemplo [31]:

```

<xs:attributeGroup name="personattr">
  <xs:attribute name="attr1" type="string"/>
  <xs:attribute name="attr2" type="integer"/>
</xs:attributeGroup>

<xs:complexType name="person">
  <xs:attributeGroup ref="personattr"/>
</xs:complexType>

```

**El grupo de atributos “personattr” contiene los atributos “attr1” y “attr2”, por lo que cualquier tipo que lo utilice (en este caso el tipo complejo “producto”), tendrá esos dos atributos implícitamente, no es necesario volver a definirlos.

Objeto “substitutionGroup”

Objeto que permite sustituir elementos por otros elementos.

Ejemplo [33]:

```

<element name="shipComment" type="string"
  substitutionGroup="ipo:comment"/>
<element name="customerComment" type="string"
  substitutionGroup="ipo:comment"/>

<items>
  <item partNum="833-AA">
    <productName>Lapis necklace</productName>

```

```

<quantity>1</quantity>
<USPrice>99.95</USPrice>
<ipo:shipComment>
  Use gold wrap if possible
</ipo:shipComment>
<ipo:customerComment>
  Want this for the holidays!
</ipo:customerComment>
<shipDate>1999-12-05</shipDate>
</item>
</items>

```

** En el ejemplo se definen dos elementos, “shipcomment”, que es el elemento cabecera, y “customerComment”, que es el atributo sustitutivo. Se puede ver que el segundo puede aparecer en lugar del primero, siempre y cuando sean del mismo tipo.

Objeto “restriction”

Objeto que permite restringir los valores de un determinado elemento.

Ejemplo [7]:

```

<xsd:element name="temperatura">
  <xsd:simpleType>
    <xsd:restriction base="xsd:decimal">
      <xsd:minInclusive value="-10"/>
      <xsd:maxInclusive value="120"/>
    </xsd:restriction>
  </xsd:simpleType>
</xsd:element>

```

Objeto “extensión”

Objeto que permite crear un tipo nuevo de secuencia a partir de un tipo de elemento “simpleType” o “complexType” existente. De esta manera no es necesario crear un nuevo elemento, si no que se puede ajustar determinadas secuencias para casos específicos. A su vez, posee los siguientes atributos:

- id: Opcional. Identifica únivocamente un elemento
- base: Obligatorio. Especifica el nombre del tipo que es ampliado
- any attributes: Especifica otros atributos sin espacio de nombres

Ejemplo [31]:

```

<xs:simpleType name="size">
  <xs:restriction base="xs:string">
    <xs:enumeration value="small" />
    <xs:enumeration value="medium" />
  </xs:restriction>
</xs:simpleType>

```

```

        <xs:enumeration value="large" />
    </xs:restriction>
</xs:simpleType>

<xs:complexType name="jeans">
    <xs:simpleContent>
        <xs:extension base="size">
            <xs:attribute name="sex">
                <xs:simpleType>
                    <xs:restriction base="xs:string">
                        <xs:enumeration value="male" />
                        <xs:enumeration value="female" />
                    </xs:restriction>
                </xs:simpleType>
            </xs:attribute>
        </xs:extension>
    </xs:simpleContent>
</xs:complexType>

```

Objeto "anyAttribute"

Objeto que indica la presencia de cualquier atributo dentro de un elemento.

Ejemplo:

```

<xsd:element name="person">
    <xsd:complexType>
        <xsd:sequence>
            <xsd:element name="name"/>
        </xsd:sequence>
        <xsd:anyAttribute/>
    </xsd:complexType>
</xsd:element>

```

Objeto "choice"

Especifica una lista de elementos alternativos para incluirse en un determinado elemento.

Ejemplo [7]:

```

<xsd:element name="ubicacion">
    <xsd:complexType>
        <xsd:choice>
            <xsd:element ref="ciudad"/>
            <xsd:element ref="municipio"/>
        </xsd:choice>
    </xsd:complexType>
</xsd:element>

```

```
</xsd:complexType>  
</xsd:element>
```

Objeto “sequence”

Objeto que especifica una lista de elementos que deberán incluirse en un determinado elemento. La lista podrá aparecer de 0 a N veces.

A su vez, tiene varios atributos propios:

- Id: Opcional. Identifica únivocamente un elemento.
- maxOccurs: Opcional. Número máximo de veces que puede repetirse la secuencia dentro del elemento padre.
- minOccurs: Opcional. Número mínimo de veces que puede repetirse la secuencia dentro del elemento padre.
- any attributes: Opcional. Especifica otros atributos sin esquema de nombres.

Ejemplo [33]:

```
<xsd:element name="personinfo">  
  <xsd:complexType>  
    <xsd:sequence>  
      <xsd:element name="firstname" type="xs:string"/>  
      <xsd:element name="lastname" type="xs:string"/>  
      <xsd:element name="address" type="xs:string"/>  
      <xsd:element name="city" type="xs:string"/>  
      <xsd:element name="country" type="xs:string"/>  
    </xsd:sequence>  
  </xsd:complexType>  
</xsd:element>
```

4.2 Tipos de Datos XML

Al igual que es necesario tener claros algunos conceptos de XML, es preciso conocer los tipos de datos que XBRL utiliza procedentes de XML.

Tipo de datos "Name"

Representa nombres en XML. Name es un símbolo que empieza con una letra, carácter de subrayado o signo de dos puntos, y continúa con caracteres de nombre (letras, dígitos y otros caracteres). Este tipo de datos se deriva de token.

Ejemplo [XML Schema Part 0 Primer Second Edition]:

"shipTo"

Tipo de datos "NCName"

Literalmente significa "non-colonized Names". Representa nombres sin el signo de dos puntos. Este tipo de datos es igual que **Name**, excepto en que no puede empezar con el signo de dos puntos. Este tipo de datos por tanto se deriva de **Name**.

Ejemplo: [13]

```
<xs:element name="OuterOne">
  <xs:complexType>
    <xs:choice maxOccurs="unbounded">
      <xs:element ref="MiddleOne"/>
      <xs:element ref="MiddleTwo"/>
    </xs:choice>
    <xs:attribute name="id" use="required" type="xs:NCName"/>
  </xs:complexType>
</xs:element>

<OuterOne id="Outer1">
  <AnotherLevel>
    <MiddleOne id="Middle1_1">
      <Inner>8</Inner>
    </MiddleOne>
    <MiddleTwo id="Middle2_1">
      <Inner>6</Inner>
    </MiddleTwo>
    <MiddleOne id="Middle1_2">
      <Inner>3</Inner>
    </MiddleOne>
  </AnotherLevel>
</OuterOne>
```



```
</MiddleOne>
<MiddleTwo id="Middle2_3">
  <Inner>1</Inner>
</MiddleTwo>
</AnotherLevel>
</OuterOne>
```

Tipo de datos "NMTOKEN"

NMTOKEN es un conjunto de caracteres de nombres (letras, dígitos y otros caracteres) en cualquier combinación. A diferencia de Name y NCName, NMTOKEN, no tiene restricciones del carácter inicial. Este tipo de datos se deriva de token.

Ejemplo [13]:

"US,Brésil"

Tipo de datos "NMTokens"

Contiene un conjunto de valores de tipo **NMTOKEN**.

Ejemplo [13]:

"US UK,
Brésil Canada Mexique"

Tipo de datos "anyURI"

Contiene el valor de un "Uniform Resource Identifier Reference".

Ejemplo [13]:

["http://www.example.com/](http://www.example.com/),
[http://www.example.com/doc.html#ID5"](http://www.example.com/doc.html#ID5)

Tipo de datos "processContent"

Indica como el lenguaje XML de otros espacios de nombres debe ser validado. Sus valores pueden ser:

- strict: Debe validar el XML con el esquema asociado al espacio de nombres indicado.
- lax: El XML debe ser validado contra su propio esquema
- skip: No se valida el XML.

Ejemplo [33]:

```

<element name="purchaseReport">
  <complexType>
    <sequence>
      <element name="regions" type="r:RegionsType"/>
      <element name="parts" type="r:PartsType"/>
      <element name="htmlExample">
        <complexType>
          <sequence>
            <any namespace="http://www.w3.org/1999/xhtml"
              minOccurs="1" maxOccurs="unbounded"
              processContents="skip"/>
          </sequence>
        </complexType>
      </element>
    </sequence>
    <attribute name="period" type="duration"/>
    <attribute name="periodEnding" type="date"/>
  </complexType>
</element>

```

Tipo de datos "QName"

Indica como el lenguaje XML de otros espacios de nombres debe ser validado. Sus valores pueden ser:

- strict: Debe validar el XML con el esquema asociado al espacio de nombres indicado.
- lax: El XML debe ser validado contra su propio esquema
- skip: No se valida el XML.

Ejemplo [33]:

"po:USAddress"

4.3 Objetos XBRL

A continuación detallaremos los objetos XBRL sobre los que se asienta el meta-metamodelo objeto del proyecto. Hemos establecido el siguiente orden a la hora de detallarlo: Atributos, Grupos de Atributos, Tipos Simples, Tipos complejos y Elementos, aunque como veremos según nos adentremos en los objetos, no existe un orden lógico que nos permita detallarlos de forma estructurada [10].

4.3.1 Atributos XBRL

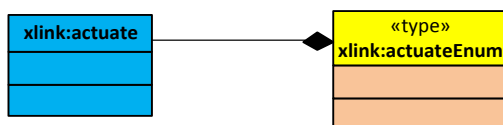
A continuación detallaremos los atributos que forman parte del meta-metamodelo de XBRL.

Atributo *xlink:actuate*

1- Definición

```
<attribute name="actuate" type="xlink:actuateEnum"/>
```

2- Modelo



3- Descripción

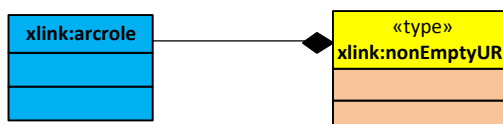
Atributo basado en el tipo [xlink:actuateEnum](#).

Atributo *xlink:arcrole*

1- Definición

```
<attribute name="arcrole" type="xlink:nonEmptyURI"/>
```

2- Modelo



3- Descripción

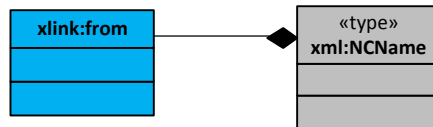
Atributo basado en el tipo [xlink:nonEmptyURI](#). Define el significado de los recursos dentro del contexto de un enlace. Proporciona información extra para que cada aplicación pueda diferenciar cada arco en cada enlace. Se utiliza para definir la relación entre dos conceptos o entre estos y recursos.

Atributo *xlink:from*

1- Definición

```
<attribute name="from" type="NCName"/>
```

2- Modelo



3- Descripción

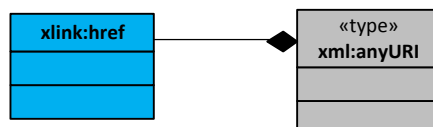
Aparece en los arcos. Su valor es el mismo que el de un atributo label, de al menos un localizador de recursos. Su valor debe ser un XML [NCName](#) (comienza por una letra o guión bajo).

Atributo *xlink:href*

1- Definición

```
<attribute name="href" type="anyURI"/>
```

2- Modelo



3- Descripción

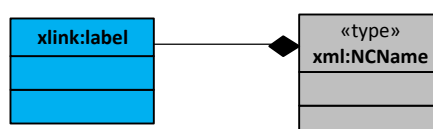
Atributo que suministra los datos que permite a la aplicación XLink encontrar los recursos remotos. Debe ser utilizado en localizadores y su valor es una [URI](#).

Atributo *xlink:label*

1- Definición

```
<attribute name="label" type="NCName"/>
```

2- Modelo



3- Descripción

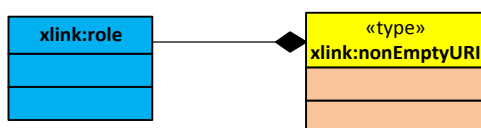
Identifica el localizador, con el objetivo de que los arcos en el mismo vínculo extendido puedan referenciarse. Su valor es un [NCName](#).

Atributo *xlink:role*

1- Definición

```
<attribute name="role" type="xlink:nonEmptyURI"/>
```

2- Modelo



3- Descripción

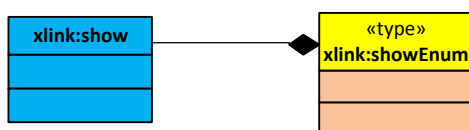
Describe el significado de los recursos dentro del contexto de un vínculo. Su valor debe ser una [URI](#). Basado en el tipo [xlink:nonEmptyURI](#).

Atributo *xlink:show*

1- Definición

```
<attribute name="show" type="xlink:showEnum"/>
```

2- Modelo



3- Descripción

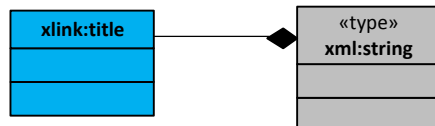
Atributo basado en el tipo [xlink:showEnum](#).

Atributo *xlink:title*

1- Definición

```
<attribute name="title" type="string"/>
```

2- Modelo



3- Descripción

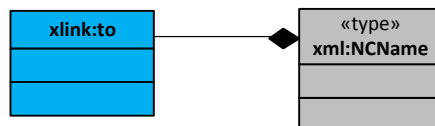
Cadena de caracteres que define un título.

Atributo xlink:to

1- Definición

```
<attribute name="to" type="NCName"/>
```

2- Modelo



3- Descripción

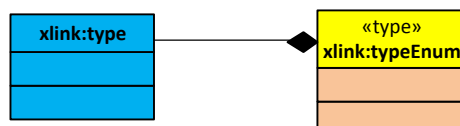
Su valor debe ser igual que el de un atributo label, de al menos un localizador o recurso en el mismo vínculo extendido. Su valor debe ser un XML [NCName](#).

Atributo xlink:type

1- Definición

```
<attribute name="type" type="xlink:typeEnum"/>
```

2- Modelo



3- Descripción

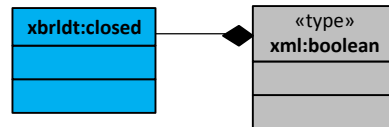
El atributo type es necesario para indicar el tipo de dato que informa el elemento. Esta basado en el atributo [xlink:typeEnum](#).

Atributo *xbrldt:closed*

1- Definición

```
<xs:attribute name="closed" type="xs:boolean" default="false"/>
```

2- Modelo



3- Descripción

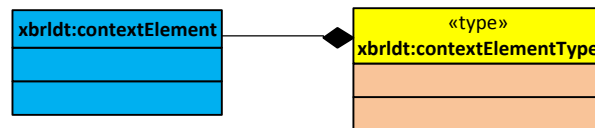
Atributo booleano (verdadero/falso).

Atributo *xbrldt:contextElement*

1- Definición

```
<xs:attribute name="contextElement" type="xbrldt:contextElementType"/>
```

2- Modelo



3- Descripción

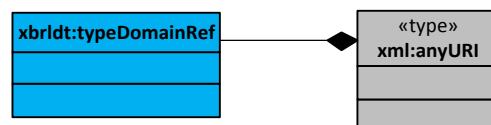
Atributo basado en el tipo [contextElementType](#), por lo que estará compuesto por segmento o un escenario.

Atributo *xbrldt:typeDomainRef*

1- Definición

```
<xs:attribute name="typedDomainRef" type="xs:anyURI"/>
```

2- Modelo



3- Descripción

Atributo basado en tipo XML “[anyURI](#)”.

4- Ejemplo

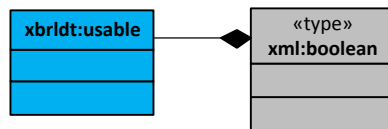
```
<xsd:element id="dim_EI" xbrli:periodType="duration"
xbrldt:typedDomainRef="../com/d/dEC/comdEC.xsd#dEC_entityCode" abstract="true" name="EI"
nillable="true" substitutionGroup="xbrldt:dimensionItem" type="xbrli:stringItemType" />
```

Atributo *xbrldt:usable*

1- Definición

```
<xs:attribute name="usable" type="xs:boolean" default="true"/>
```

2- Modelo



3- Descripción

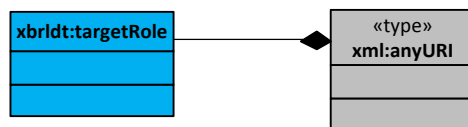
Atributo booleano.

Atributo *xbrldt:targetRole*

1- Definición

```
<xs:attribute name="targetRole" type="xs:anyURI"/>
```

2- Modelo



3- Descripción

Atributo basado en el tipo XML “[anyURI](#)”.

Atributo *xbrli:balance*

1- Definición

```
<attribute name="balance">
```

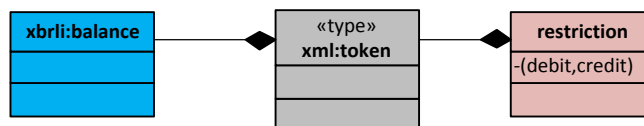


```

<annotation>
  <documentation>
    The balance attribute (imposes calculation relationship restrictions)
  </documentation>
</annotation>
<simpleType>
  <restriction base="token">
    <enumeration value="debit" />
    <enumeration value="credit" />
  </restriction>
</simpleType>
</attribute>

```

2- Modelo



3- Descripción

Atributo opcional que aparece normalmente en los ítems cuyo tipo es [monetaryItemType](#). Es un token y puede tomar los valores “credit” o “debit”. Sirve para mejorar la validación de cálculos, en cuanto a saldos deudores y acreedores se refiere. Sólo tiene lugar con valores de tipo monetario o derivados de él.

4- Ejemplo [8]:

```

<xsd:element name="AssetsCurrentTotal" id="ifrs-gp_AssetsCurrentTotal"
  type="xbrli:monetaryItemType"
  substitutionGroup="xbrli:item" nillable="true"
  xbrli:balance="debit"
  xbrli:periodType="instant"/>

```

Atributo *xbrli:periodType*

1- Definición

```

<attribute name="periodType">
  <annotation>
    <documentation>
      The periodType attribute (restricting the period for XBRL items)
    </documentation>
  </annotation>
  <simpleType>
    <restriction base="token">
      <enumeration value="instant" />
    </restriction>
  </simpleType>
</attribute>

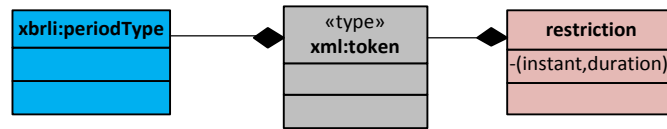
```

```

    <enumeration value="duration" />
  </restriction>
</simpleType>
</attribute>

```

2- Modelo



3- Descripción

Atributo que permite identificar la relación que el concepto tiene con la coordenada tiempo. El valor instant indica que el elemento que se utilice en una instancia XBRL deberá asociarse a un contexto en el que el periodo es un instante. El valor duración se asocia a un rango de fechas en las cuales el dato es válido.

4- Ejemplo [8]:

```

<xsd:element name="AssetsCurrentTotal" id="ifrs-gp_AssetsCurrentTotal"
  type="xbrli:monetaryItemType"
  substitutionGroup="xbrli:item" nillable="true"
  xbrli:balance="debit"
  xbrli:periodType="instant"/>

```

4.3.2 Grupos de Atributos XBRL

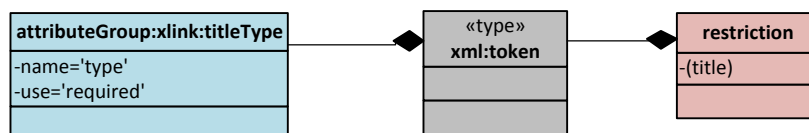
A continuación detallaremos los grupos de atributos que forman parte del meta-metamodelo de XBRL.

Grupo *xlink:titleType*

1- Definición

```
<attributeGroup name="titleType">
  <attribute name="type" use="required">
    <simpleType>
      <restriction base="token">
        <enumeration value="title"/>
      </restriction>
    </simpleType>
  </attribute>
</attributeGroup>
```

2- Modelo



3- Descripción

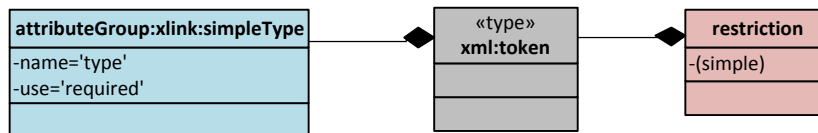
Grupo de atributos donde se define el atributo “[type](#)” y el token “title”.

Grupo *xlink:simpleType*

1- Definición

```
<attributeGroup name="simpleType">
  <attribute name="type" use="required">
    <simpleType>
      <restriction base="token">
        <enumeration value="simple"/>
      </restriction>
    </simpleType>
  </attribute>
</attributeGroup>
```

2- Modelo



3- Descripción

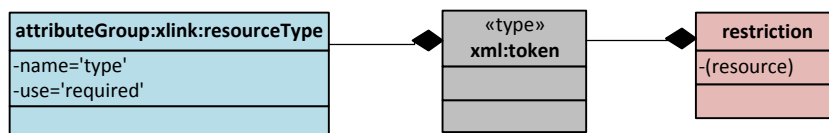
Grupo de atributos donde se define el atributo “[type](#)” y el token “simple”.

Grupo *xlink:resourceType*

1- Definición

```
<attributeGroup name="resourceType">
  <attribute name="type" use="required">
    <simpleType>
      <restriction base="token">
        <enumeration value="resource"/>
      </restriction>
    </simpleType>
  </attribute>
</attributeGroup>
```

2- Modelo



3- Descripción

Grupo de atributos donde se define el atributo “[type](#)” y el token “resource”.

Grupo *xlink:locatorType*

1- Definición

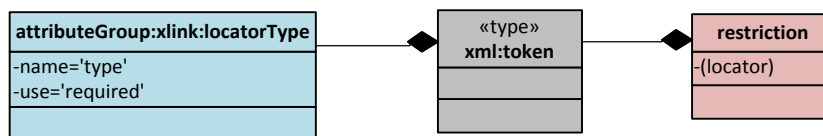
```
<attributeGroup name="locatorType">
  <attribute name="type" use="required">
    <simpleType>
      <restriction base="token">
        <enumeration value="locator"/>
      </restriction>
    </simpleType>
  </attribute>
</attributeGroup>
```

```

</attribute>
</attributeGroup>

```

2- Modelo



3- Descripción

Grupo de atributos donde se define el atributo “[type](#)” y el valor “locator”.

Grupo xlink:extendedType

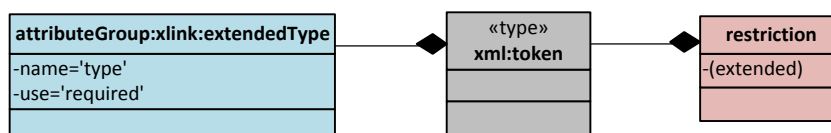
1- Definición

```

<attributeGroup name="extendedType">
  <attribute name="type" use="required">
    <simpleType>
      <restriction base="token">
        <enumeration value="extended"/>
      </restriction>
    </simpleType>
  </attribute>
</attributeGroup>

```

2- Modelo



3- Descripción

Grupo de atributos donde se define el atributo “[type](#)” y el valor “extended”.

Grupo xlink:arcType

1- Definición

```

<attributeGroup name="arcType">
  <attribute name="type" use="required">
    <simpleType>
      <restriction base="token">

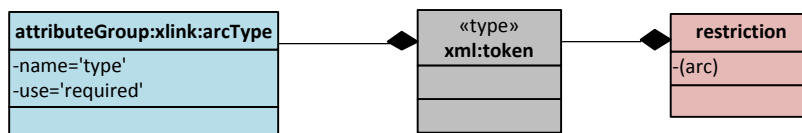
```

```

    <enumeration value="arc"/>
  </restriction>
</simpleType>
</attribute>
</attributeGroup>

```

2- Modelo



3- Descripción

Grupo de atributos donde se define el atributo “[type](#)” y el valor “arc”.

Grupo xbrli:factAttrs

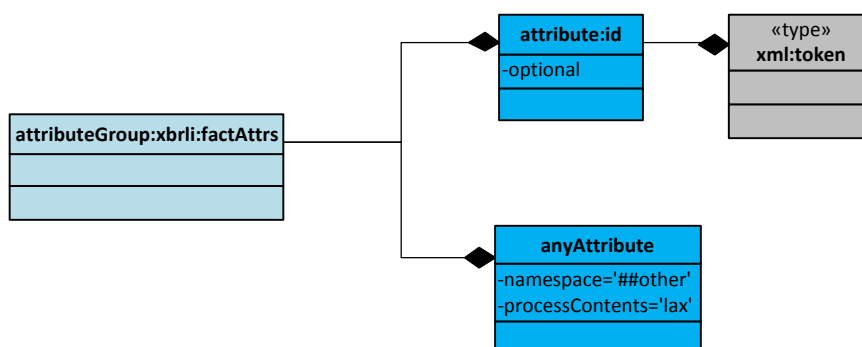
1- Definición

```

<attributeGroup name="factAttrs">
  <annotation>
    <documentation>
      Attributes for all items and tuples.
    </documentation>
  </annotation>
  <attribute name="id" type="ID" use="optional" />
  <anyAttribute namespace="##other" processContents="lax" />
</attributeGroup>

```

2- Modelo



3- Descripción

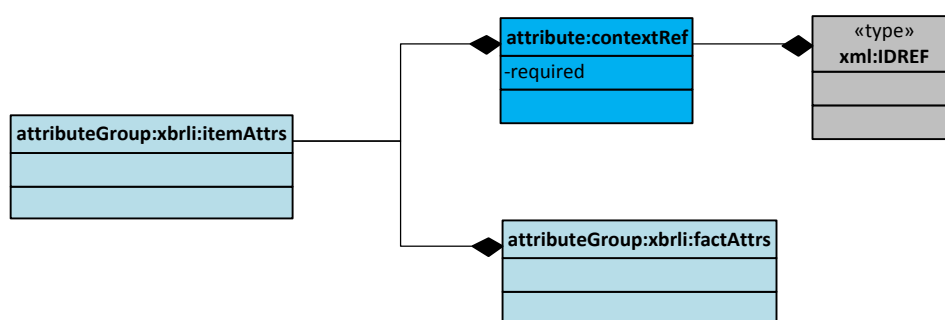
Grupo de atributos definido para cualquier elemento “[item](#)” o “[tuples](#)”.

Grupo *xbri:itemAttrs*

1- Definición

```
<attributeGroup name="itemAttrs">
  <annotation>
    <documentation>
      Attributes for all items.
    </documentation>
  </annotation>
  <attributeGroup ref="xbri:factAttrs" />
  <attribute name="contextRef" type="IDREF" use="required" />
</attributeGroup>
```

2- Modelo



3- Descripción

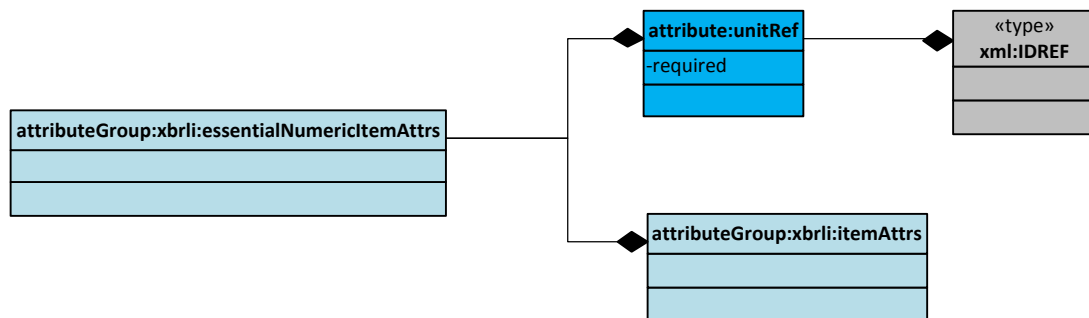
Grupo de atributos definido para cualquier elemento [“item”](#).

Grupo *xbri:essentialNumericItemAttrs*

1- Definición

```
<attributeGroup name="essentialNumericItemAttrs">
  <annotation>
    <documentation>
      Attributes for all numeric items (fractional and non-fractional).
    </documentation>
  </annotation>
  <attributeGroup ref="xbri:itemAttrs" />
  <attribute name="unitRef" type="IDREF" use="required" />
</attributeGroup>
```

2- Modelo



3- Descripción

Grupo de atributos definido para cualquier elemento [numérico](#).

Grupo xbrli:nonNumericItemAttrs

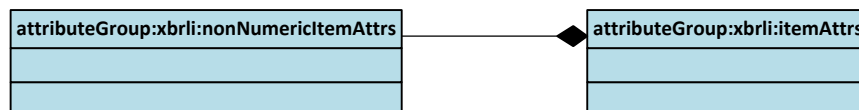
1- Definición

```

<attributeGroup name="nonNumericItemAttrs">
  <annotation>
    <documentation>
      Group of attributes for non-numeric items
    </documentation>
  </annotation>
  <attributeGroup ref="xbrli:itemAttrs" />
</attributeGroup>

```

2- Modelo



3- Descripción

Grupo de atributos definido para cualquier elemento [no numérico](#).

Grupo xbrli:numericItemAttrs

1- Definición

```

<attributeGroup name="numericItemAttrs">
  <annotation>
    <documentation>
      Group of attributes for non-fractional numeric items
    </documentation>
  </annotation>

```

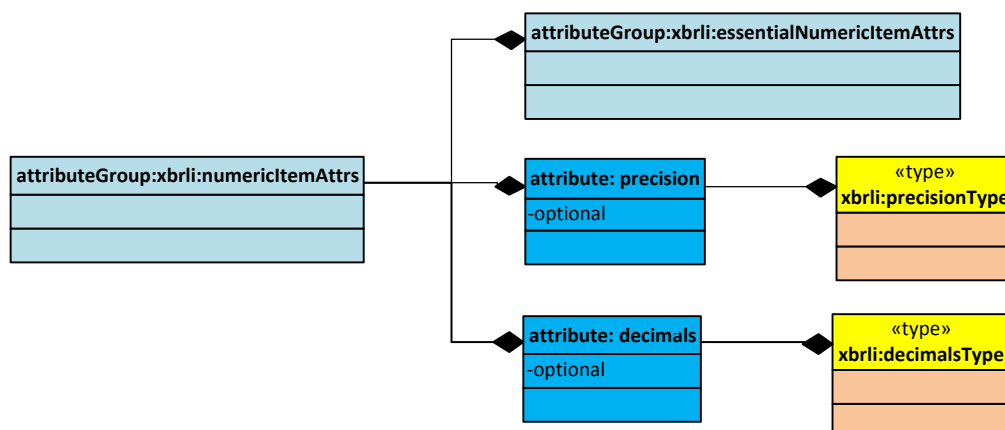


```

</annotation>
<attributeGroup ref="xbrli:essentialNumericItemAttrs" />
<attribute name="precision" type="xbrli:precisionType" use="optional" />
<attribute name="decimals" type="xbrli:decimalsType" use="optional" />
</attributeGroup>

```

2- Modelo



3- Descripción

Grupo de atributos definido para cualquier elemento numérico no fraccionado.

El grupo de atributos “numericItemAttrs” está formado por:

- El grupo “[essentialNumericItemAttrs](#)”
- Los atributos “precisión” y “decimals”, de tipo “[precisionType](#)” y “[decimalsType](#)” respectivamente.

Grupo xbrli:tupleAttrs

1- Definición

```

<attributeGroup name="tupleAttrs">
  <annotation>
    <documentation>
      Group of attributes for tuples.
    </documentation>
  </annotation>
  <attributeGroup ref="xbrli:factAttrs" />
</attributeGroup>

```

2- Modelo



3- Descripción

Grupo de atributos definido para cualquier elemento “[tuples](#)”.

4.3.3 Tipos Simples XBRL

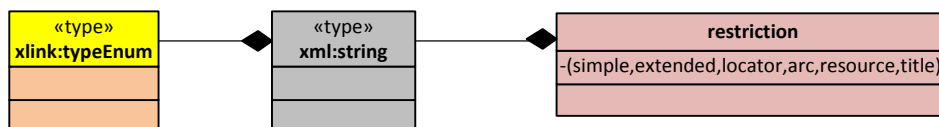
A continuación detallaremos los tipos simples que forman parte del meta-metamodelo de XBRL.

Tipo xlink:typeEnum

1- Definición

```
<simpleType name="typeEnum">
  <annotation>
    <documentation>
      Enumeration of values for the type attribute
    </documentation>
  </annotation>
  <restriction base="string">
    <enumeration value="simple"/>
    <enumeration value="extended"/>
    <enumeration value="locator"/>
    <enumeration value="arc"/>
    <enumeration value="resource"/>
    <enumeration value="title"/>
  </restriction>
</simpleType>
```

2- Modelo



3- Descripción

Enumera los posibles valores que puede tomar el atributo [“type”](#).

Tipo xlink:nonEmptyURI

1- Definición

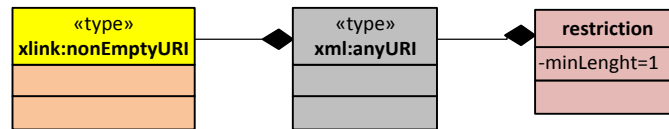
```
<simpleType name="nonEmptyURI">
  <annotation>
    <documentation>
      A URI type with a minimum length of 1 character.
      Used on role and arcrole and href elements.
    </documentation>
  </annotation>
  <restriction base="anyURI">
```

```

    <minLength value="1"/>
  </restriction>
</simpleType>

```

2- Modelo



3- Descripción

Este tipo tiene como objetivo el uso de los URIs en XBRL. Al atributo minLength indica que la longitud mínima es de un carácter, es decir, no puede ser una cadena vacía.

Tipo xlink:actuateEnum

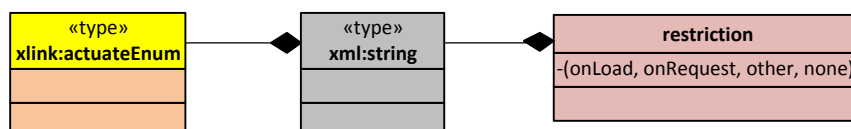
1- Definición

```

<simpleType name="actuateEnum">
  <annotation>
    <documentation>
      Enumeration of values for the actuate attribute
    </documentation>
  </annotation>
  <restriction base="string">
    <enumeration value="onLoad"/>
    <enumeration value="onRequest"/>
    <enumeration value="other"/>
    <enumeration value="none"/>
  </restriction>
</simpleType>

```

2- Modelo



3- Descripción

Señala los posibles valores que puede tomar el atributo “[actuate](#)”, y que indican el momento de activar el enlace. Estos valores son:

- “onLoad” → El enlace se activa en el momento de la carga del documento.

- “onRequest” → El enlace se activa durante la activación del elemento. “other” → Otra opción no determinada.
- “none” → Ninguna opción especificada.

Tipo xlink:showEnum

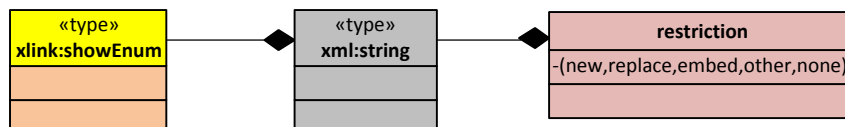
1- Definición

```

<simpleType name="showEnum">
  <annotation>
    <documentation>
      Enumeration of values for the show attribute
    </documentation>
  </annotation>
  <restriction base="string">
    <enumeration value="new"/>
    <enumeration value="replace"/>
    <enumeration value="embed"/>
    <enumeration value="other"/>
    <enumeration value="none"/>
  </restriction>
</simpleType>

```

2- Modelo



3- Descripción

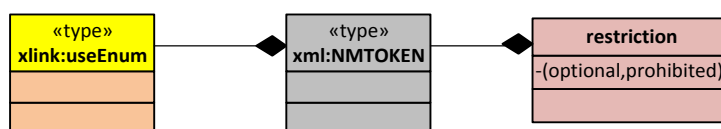
Especifica los posibles valores que puede tomar el atributo ‘[show](#)’, y que indica la forma en que se carga/muestra el recurso. Son:

- “new” → El objeto se carga en un nuevo elemento (por ejemplo, un link que se abre en una nueva ventana).
- “replace” → El objeto se carga en el mismo elemento actual (por ejemplo, un link que se abre en la ventana actual).
- “embed” → Se carga una presentación del objeto en lugar del objeto en sí.
- “other” → Otra opción no determinada.
- “none” → Ninguna opción especificada.

1- Definición

```
<simpleType name="useEnum">
  <annotation>
    <documentation>
      Enumerated values for the use attribute on extended link arcs.
    </documentation>
  </annotation>
  <restriction base="NMTOKEN">
    <enumeration value="optional" />
    <enumeration value="prohibited" />
  </restriction>
</simpleType>
```

2- Modelo



3- Descripción

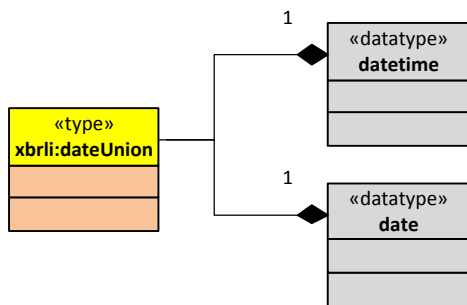
Indica los posibles valores del atributo “use” en los arcos, que son:

- “optional” → Expresa una relación que podría participar en la red de relaciones definidas en un DTS.
- “prohibited” → La relación no existe.

1- Definición

```
<simpleType name="dateUnion">
  <annotation>
    <documentation>
      The union of the date and dateTime simple types.
    </documentation>
  </annotation>
  <union memberTypes="date dateTime" />
</simpleType>
```

2- Modelo



3- Descripción

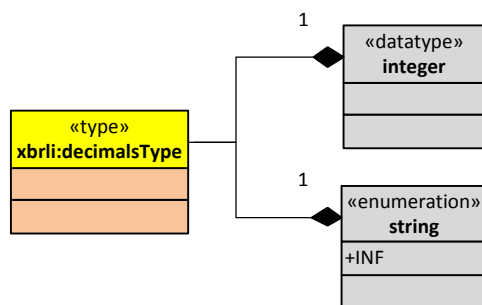
Unión de los tipos XML “datetime” y “date”. Como resultado tendremos un tipo con la fecha y la hora.

Tipo xbrli:decimalsType

1- Definición

```
<simpleType name="decimalsType">
  <annotation>
    <documentation>
      This type is used to specify the value of the decimals attribute
      on numeric items. It consists of the union of integer and "INF"
      (used to signify that a number is expressed to an infinite number
      of decimal places or "exact value").
    </documentation>
  </annotation>
  <union memberTypes="integer ">
    <simpleType>
      <restriction base="string">
        <enumeration value="INF" />
      </restriction>
    </simpleType>
  </union>
</simpleType>
```

2- Modelo



3- Descripción

Define los valores que puede tomar el atributo “decimals”. Esto es el número de cifras decimales que puede tener un valor de un hecho. Estos pueden ser un entero, o la cadena “INF”, que indica que el valor representado es el valor exacto del hecho. Si el valor es positivo, indica el número de decimales significativos. Si es negativo, indica que el valor es conocido con exactitud hasta el dígito que ocupa la posición a la izquierda del punto decimal que expresa “decimals”, éste inclusive. Si toma el valor “INF”, es que el valor es conocido con total exactitud, y si es “0”, el número de decimales significativos es desconocido.

4- Ejemplo [8]:

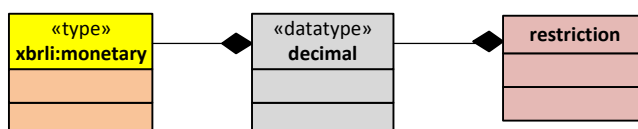
```
<ifrs-gp:AssetsCurrentTotal decimals="0" contextRef="PERIODO2007"
unitRef="USD">18000</ifrs-gp:AssetsCurrentTotal>
```

Tipo xbrli:monetary

1- Definición

```
<simpleType name="monetary">
  <annotation>
    <documentation>
      the monetary type serves as the datatype for those financial
      concepts in a taxonomy which denote units in a currency.
      Instance items with this type must have a unit of measure
      from the ISO 4217 namespace of currencies.
    </documentation>
  </annotation>
  <restriction base="decimal" />
</simpleType>
```

2- Modelo



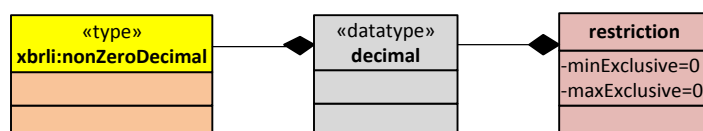
3- Descripción

Expresa conceptos cuyas unidades son de carácter monetario, regidos por el estándar ISO 4217.

1- Definición

```
<simpleType name="nonZeroDecimal">
  <annotation>
    <documentation>
      As the name implies this is a decimal value that can not take
      the value 0 - it is used as the type for the denominator of a
      fractionItemType.
    </documentation>
  </annotation>
  <union>
    <simpleType>
      <restriction base="decimal">
        <minExclusive value="0" />
      </restriction>
    </simpleType>
    <simpleType>
      <restriction base="decimal">
        <maxExclusive value="0" />
      </restriction>
    </simpleType>
  </union>
</simpleType>
```

2- Modelo



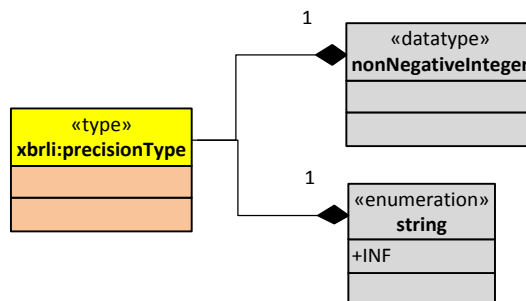
3- Descripción

Valor decimal distinto de cero y que puede ser utilizado para el denominador del tipo [xbrli:fractionItemType](#).

1- Definición

```
<simpleType name="precisionType">  
  <annotation>  
    <documentation>  
      This type is used to specify the value of the  
      precision attribute on numeric items. It consists  
      of the union of nonNegativeInteger and "INF" (used  
      to signify infinite precision or "exact value").  
    </documentation>  
  </annotation>  
  <union memberTypes="nonNegativeInteger">  
    <simpleType>  
      <restriction base="string">  
        <enumeration value="INF" />  
      </restriction>  
    </simpleType>  
  </union>  
</simpleType>
```

2- Modelo



3- Descripción

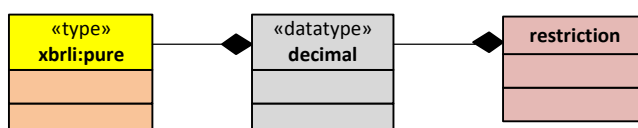
Define el espacio de valores que puede tomar el atributo “precision”. Este debe ser un entero no negativo o la cadena ‘INF’, e indica una precisión aritmética de la medida que representa (comenzando por el primer número distinto de cero de la izquierda).

Tipo xbrli:pure

1- Definición

```
<simpleType name="pure">  
  <annotation>  
    <documentation>  
      This datatype serves as the type for dimensionless numbers  
      such as percentage change, growth rates, and other ratios  
      where the numerator and denominator have the same units.  
    </documentation>  
  </annotation>  
  <restriction base="decimal" />  
</simpleType>
```

2- Modelo



3- Descripción

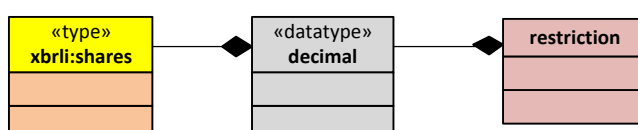
Dato numérico que no tiene ninguna unidad asociada.

Tipo xbrli:shares

1- Definición

```
<simpleType name="shares">  
  <annotation>  
    <documentation>  
      This datatype serves as the datatype for share based  
      financial concepts.  
    </documentation>  
  </annotation>  
  <restriction base="decimal" />  
</simpleType>
```

2- Modelo



3- Descripción

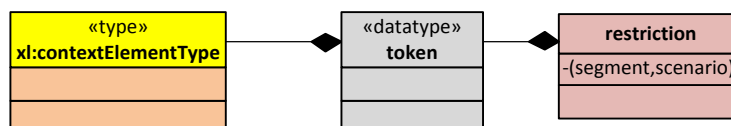
Acción o participación en el contexto financiero.

Tipo `xbrldt:contextElementType`

1- Definición

```
<xs:simpleType name="contextElementType">
  <xs:restriction base="xs:token">
    <xs:enumeration value="segment"/>
    <xs:enumeration value="scenario"/>
  </xs:restriction>
</xs:simpleType>
```

2- Modelo



3- Descripción

Indica los valores que puede tomar el atributo “`contextElement`”. Estos especifican el contexto en el que se encuadra el informe XBRL. Podrá ser un [segmento](#) o un [escenario](#). El segmento indica la dimensión de lógica de negocio donde se encuadra el contexto. El escenario indica un conjunto de dimensiones.

4- Ejemplo [28]:

```
<xbrli:context id="Context_Instant_GrupoConsolidableEntidadesCredito">
  <xbrli:entity>
    <xbrli:identifier scheme="http://www.ecb.int/stats/money/mfi">ES9000</xbrli:identifier>
    <xbrli:segment>
      <xbrldi:explicitMember dimension="es-be-cm-dim:Agrupacion">es-be-cm-
dim:AgrupacionIndividual</xbrldi:explicitMember>
    </xbrli:segment>
  </xbrli:entity>
  <xbrli:period>
    <xbrli:instant>2008-09-30</xbrli:instant>
  </xbrli:period>
  <xbrli:scenario>
    <xbrldi:explicitMember dimension="es-be-d-FR-dist:DistribucionDimension">es-be-d-FR-
dist:GrupoConsolidableEntidadesCredito</xbrldi:explicitMember>
  </xbrli:scenario>
</xbrli:context>
```

En el ejemplo podemos ver que se define el segmento como la dimensión “es-be-cm-dim:Agrupacion” y tenemos que el atributo de dimensión es “es-be-ce-dim:AgrupacionIndividual”. Además, que el periodo definido tiene su fecha de comienzo en 2008-07-01 y su fecha de final en 2008-09-30.

4.3.4 Tipos Complejos XBRL

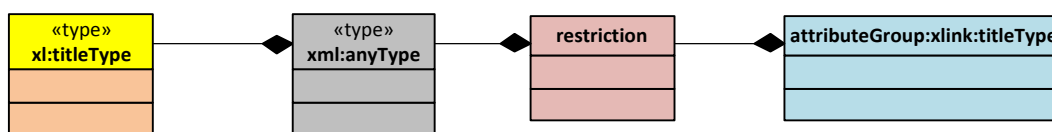
A continuación detallaremos los tipos complejos que forman parte del meta-metamodelo de XBRL.

Tipo `xl:titleType`

1- Definición

```
<complexType name="titleType">
  <annotation>
    <documentation>
      Type for the abstract title element - used as a title element template.
    </documentation>
  </annotation>
  <complexContent>
    <restriction base="anyType">
      <attributeGroup ref="xlink:titleType"/>
    </restriction>
  </complexContent>
</complexType>
```

2- Modelo



3- Descripción

Tipo basado en el tipo xml “anyType”, restringido al grupo de atributos “[titleType](#)”.

Tipo `xl:simpleType`

1- Definición

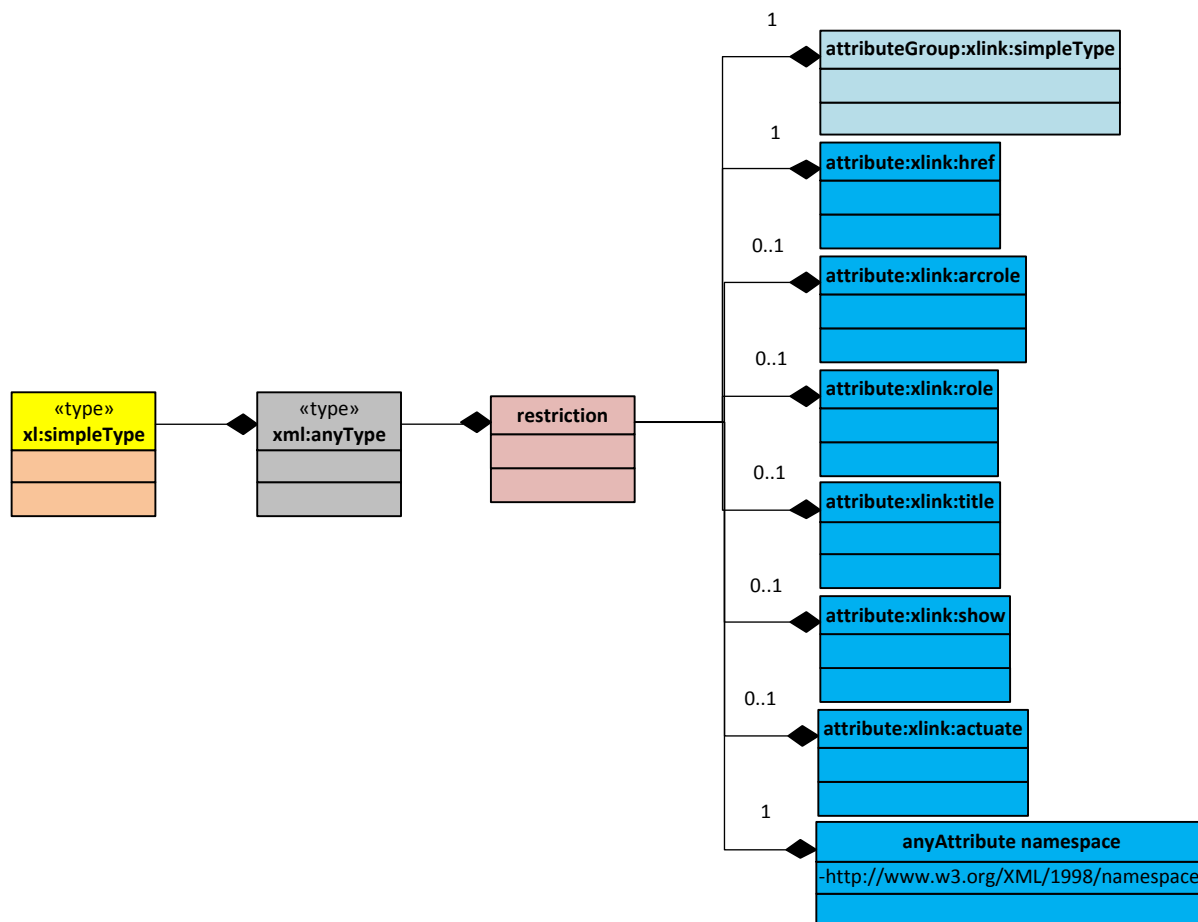
```
<complexType name="simpleType">
  <annotation>
    <documentation>
      Type for the simple links defined in XBRL
    </documentation>
  </annotation>
  <complexContent>
    <restriction base="anyType">
      <attributeGroup ref="xlink:simpleType"/>
      <attribute ref="xlink:href" use="required" />
      <attribute ref="xlink:arcrole" use="optional" />
    </restriction>
  </complexContent>
</complexType>
```

```

<attribute ref="xlink:role" use="optional" />
<attribute ref="xlink:title" use="optional" />
<attribute ref="xlink:show" use="optional" />
<attribute ref="xlink:actuate" use="optional" />
<anyAttribute namespace=http://www.w3.org/XML/1998/namespace processContents="lax"/>
</restriction>
</complexContent>
</complexType>

```

2- Modelo



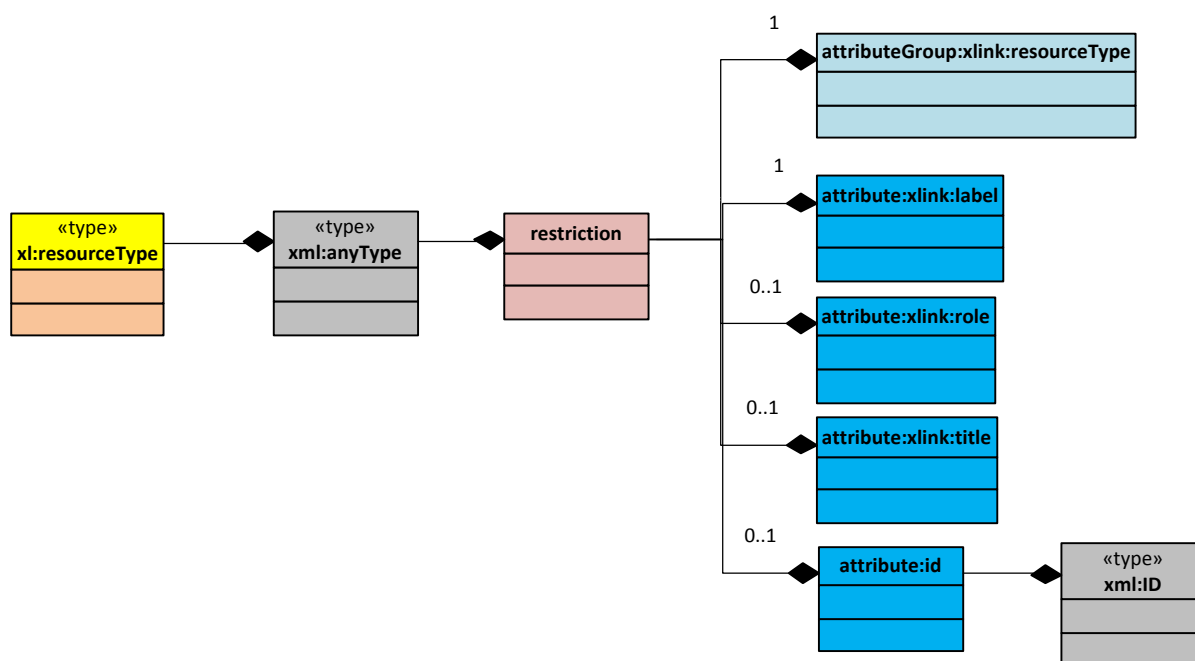
3- Descripción

Tipo basado en el tipo xml “anyType”, restringido al grupo de atributos “[simpleType](#)”, con el atributo “[href](#)” y el espacio de nombres “www.w3.org/XML/1998/namespace” como obligatorios, y los atributos xlink “[arcrole](#)”, “[role](#)”, “[title](#)”, “[show](#)”, “[actuate](#)”, como opcionales.

1- Definición

```
<complexType name="resourceType">
  <annotation>
    <documentation>
      Generic type for the resource type element
    </documentation>
  </annotation>
  <complexContent mixed="true">
    <restriction base="anyType">
      <attributeGroup ref="xlink:resourceType"/>
      <attribute ref="xlink:label" use="required" />
      <attribute ref="xlink:role" use="optional" />
      <attribute ref="xlink:title" use="optional" />
      <attribute name="id" type="ID" use="optional" />
    </restriction>
  </complexContent>
</complexType>
```

2- Modelo



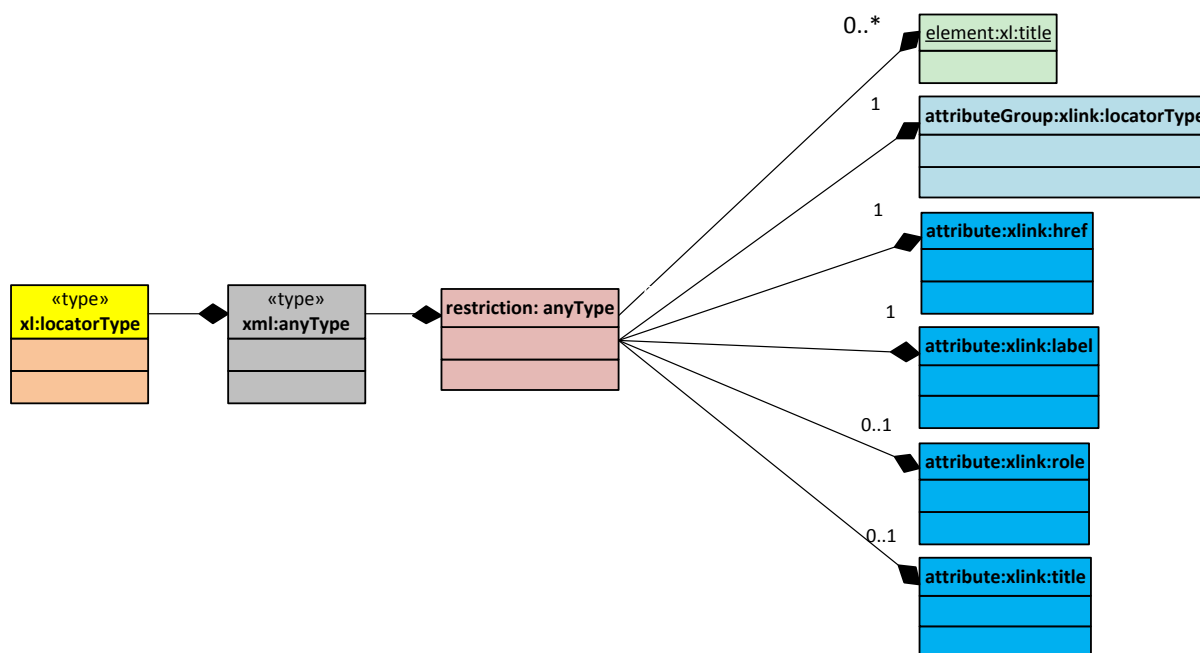
3- Descripción

Tipo basado en el tipo xml “anyType”, restringido al grupo de atributos “[resourceType](#)”, con el atributo “[label](#)” como obligatorio, y los atributos “[role](#)”, “[title](#)” e “id” (XML).

1- Definición

```
<complexType name="locatorType">
  <annotation>
    <documentation>
      Generic locator type.
    </documentation>
  </annotation>
  <complexContent>
    <restriction base="anyType">
      <sequence>
        <element ref="xl:title" minOccurs="0" maxOccurs="unbounded" />
      </sequence>
      <attributeGroup ref="xlink:locatorType"/>
      <attribute ref="xlink:href" use="required" />
      <attribute ref="xlink:label" use="required" />
      <attribute ref="xlink:role" use="optional" />
      <attribute ref="xlink:title" use="optional" />
    </restriction>
  </complexContent>
</complexType>
```

2- Modelo



3- Descripción

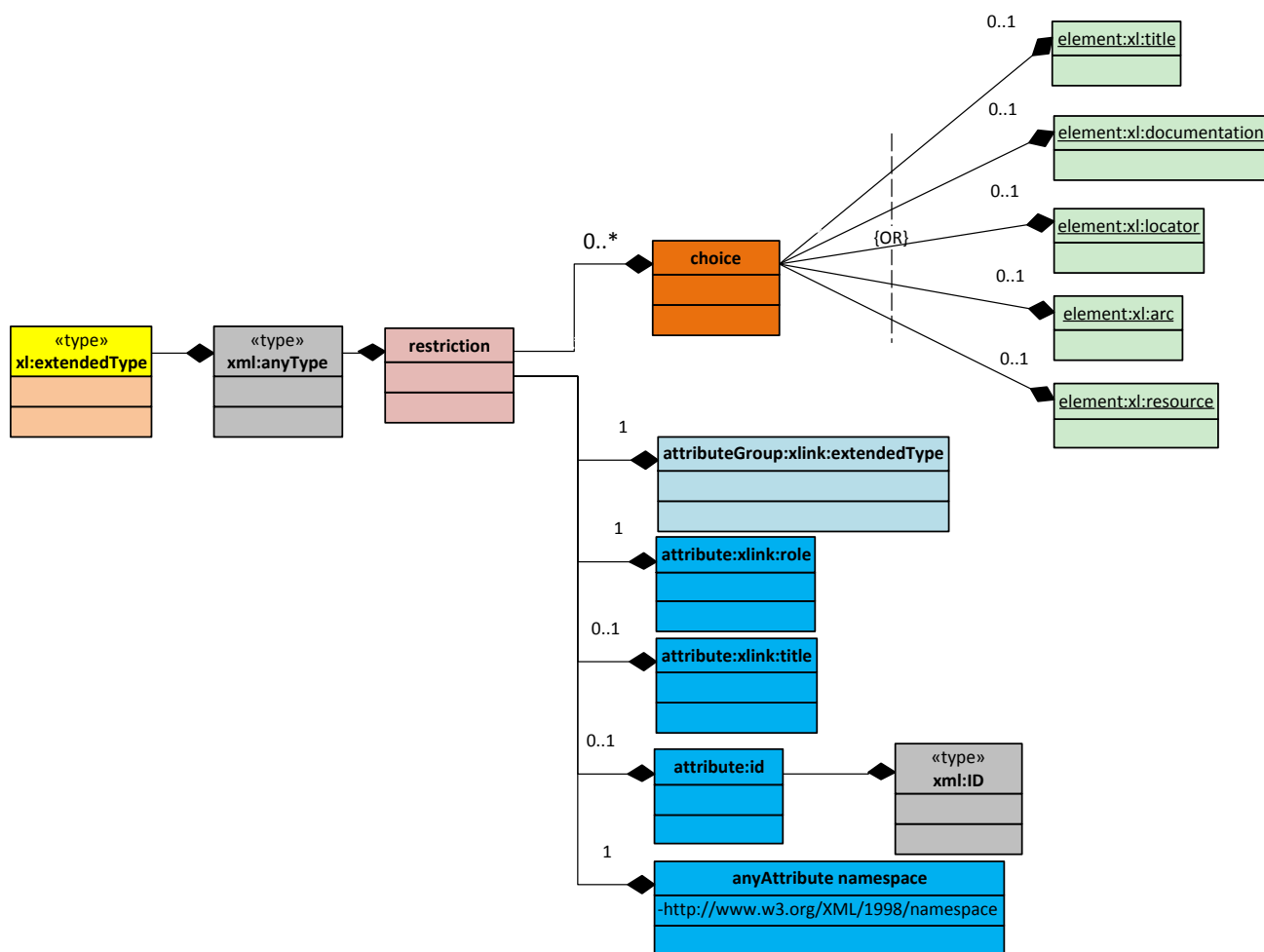
Tipo basado en el tipo xml “anyType”, restringido al elemento “title” (de 0 a N ocurrencias), al grupo de atributos “[locatorType](#)”, con los atributos “[href](#)” y “[label](#)” como obligatorio, y los atributos “[role](#)” y “[title](#)” como opcionales.

Tipo [xl:extendedType](#)

1- Definición

```
<complexType name="extendedType">
  <annotation>
    <documentation>
      Generic extended link type
    </documentation>
  </annotation>
  <complexContent>
    <restriction base="anyType">
      <choice minOccurs="0" maxOccurs="unbounded">
        <element ref="xl:title" />
        <element ref="xl:documentation" />
        <element ref="xl:locator" />
        <element ref="xl:arc" />
        <element ref="xl:resource" />
      </choice>
      <attributeGroup ref="xlink:extendedType"/>
      <attribute ref="xlink:role" use="required" />
      <attribute ref="xlink:title" use="optional" />
      <attribute name="id" type="ID" use="optional" />
      <anyAttribute namespace="http://www.w3.org/XML/1998/namespace"
processContents="lax"/>
    </restriction>
  </complexContent>
</complexType>
```

2- Modelo



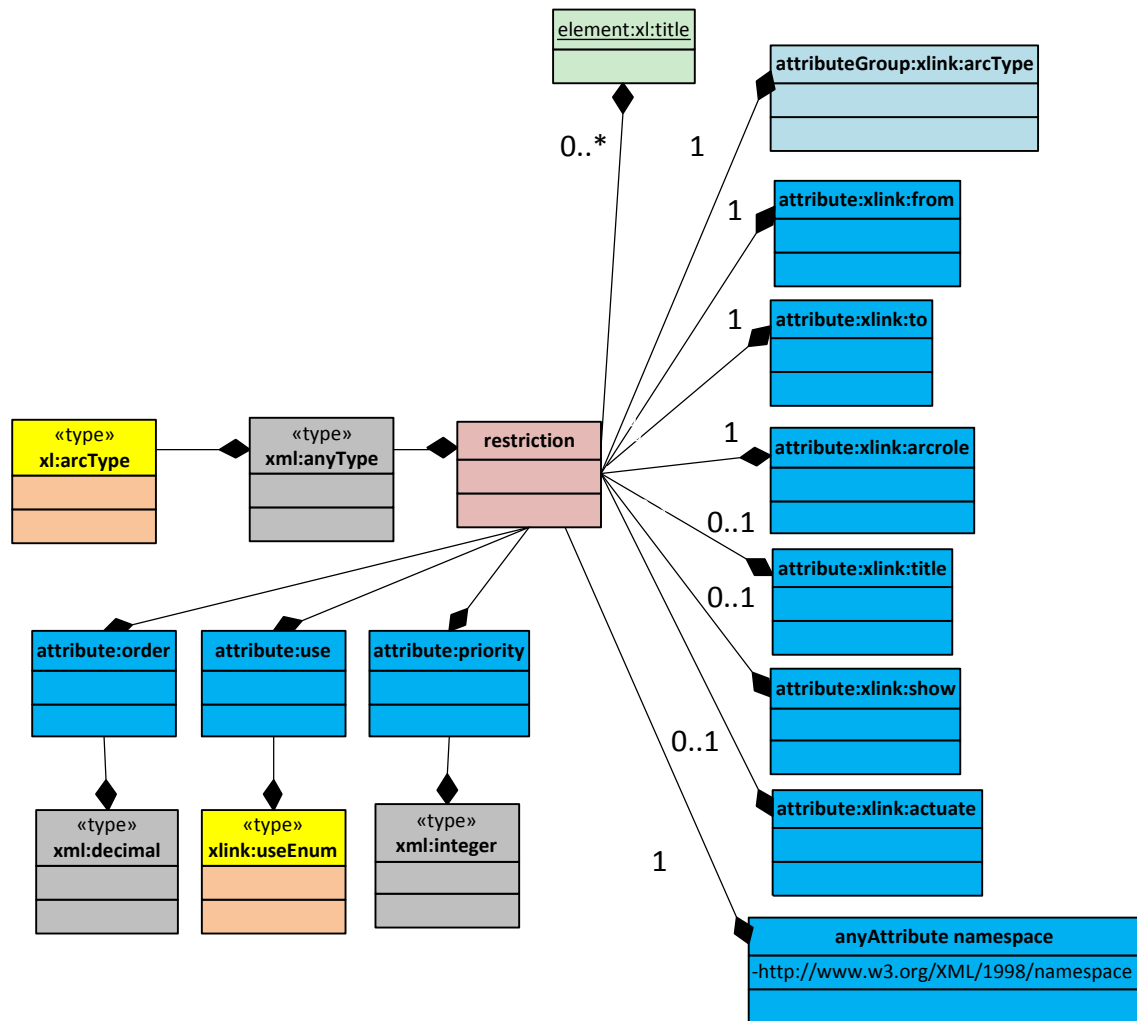
3- Descripción

Tipo basado en el tipo xml “anyType”, restringido a una secuencia de elementos (“[title](#)” o “[documentation](#)” o “[locator](#)” o “[arc](#)” o “[resource](#)”), al grupo de atributos “[extendedType](#)”, con el atributo “[role](#)” y el espacio de nombres “[www.w3.org/XML/1998/namespace](#)” como obligatorios, y los atributos “[title](#)” e “[id](#)” (XML). Son links extendidos, en fragmentos XML, donde se documenta la relación entre recursos XML.

1- Definición

```
<complexType name="arcType">
  <annotation>
    <documentation>
      basic extended link arc type - extended where necessary for specific arcs
      Extends the generic arc type by adding use, priority and order attributes.
    </documentation>
  </annotation>
  <complexContent>
    <restriction base="anyType">
      <sequence>
        <element ref="xl:title" minOccurs="0" maxOccurs="unbounded" />
      </sequence>
      <attributeGroup ref="xlink:arcType"/>
      <attribute ref="xlink:from" use="required" />
      <attribute ref="xlink:to" use="required" />
      <attribute ref="xlink:arcrole" use="required" />
      <attribute ref="xlink:title" use="optional" />
      <attribute ref="xlink:show" use="optional" />
      <attribute ref="xlink:actuate" use="optional" />
      <attribute name="order" type="decimal" use="optional" />
      <attribute name="use" type="xl:useEnum" use="optional" />
      <attribute name="priority" type="integer" use="optional" />
      <anyAttribute namespace="##other" processContents="lax" />
    </restriction>
  </complexContent>
</complexType>
```

2- Modelo



3- Descripción

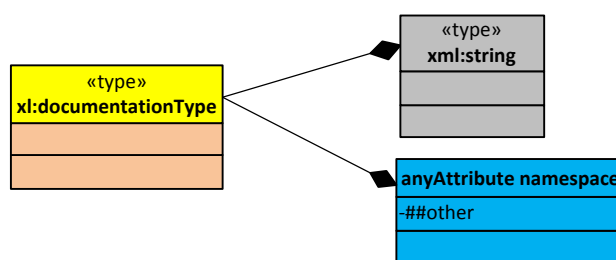
Un arco es una relación binaria entre los recursos involucrados en una relación de conceptos. Los arcos, relacionan conceptos entre sí, mediante la asociación de sus localizadores. A su vez, vinculan los recursos con los conceptos, conectando los localizadores de ambos. También se utilizan para conectar localizadores de hechos a recursos en notas al pie mediante link extendidos. Dicho de otra forma, permiten establecer relaciones entre recursos identificador por localizadores en links extendidos.

Los arcos tienen una serie de atributos (`'from'`, `'to'`, `'arcrole'`, `'title'`, `'show'` y `'actuate'`) que informan de la naturaleza de la relación, tanto de links simples como de los extendidos. Adicionalmente, tiene otros tres atributos: `"order"`, `"priority"` y `"use"`, los dos primeros basados en sus tipos XML correspondientes. El tercero, basado en el tipo `"useEnum"`.

1- Definición

```
<complexType name="documentationType">
  <annotation>
    <documentation>
      Element type to use for documentation of
      extended links and linkbases.
    </documentation>
  </annotation>
  <simpleContent>
    <extension base="string">
      <anyAttribute namespace="##other" processContents="lax"/>
    </extension>
  </simpleContent>
</complexType>
```

2- Modelo



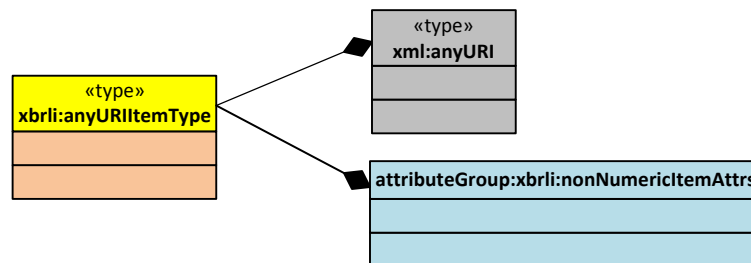
3- Descripción

Sin tener ninguna semántica específica, tiene como objetivo documentar los elementos enlace XBRL. Suele incorporar el atributo “xml:lang”, para indicar el idioma empleado. Su contenido es una cadena de texto

1- Definición

```
<complexType name="anyURIItemType" final="extension">
  <simpleContent>
    <extension base="anyURI">
      <attributeGroup ref="xbrli:nonNumericItemAttrs" />
    </extension>
  </simpleContent>
</complexType>
```

2- Modelo



3- Descripción

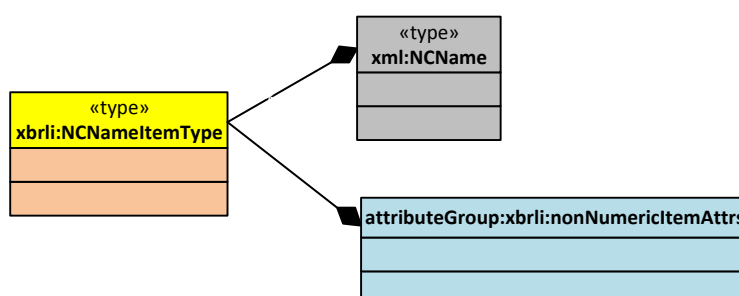
Tipo [no numérico](#) basado en el tipo XML “[anyURI](#)”, extendido con el grupo de atributos correspondiente.

Tipo *xbrli:NCNameItemType*

1- Definición

```
<complexType name="NCNameItemType" final="extension">
  <simpleContent>
    <extension base="NCName">
      <attributeGroup ref="g" />
    </extension>
  </simpleContent>
</complexType>
```

2- Modelo



3- Descripción

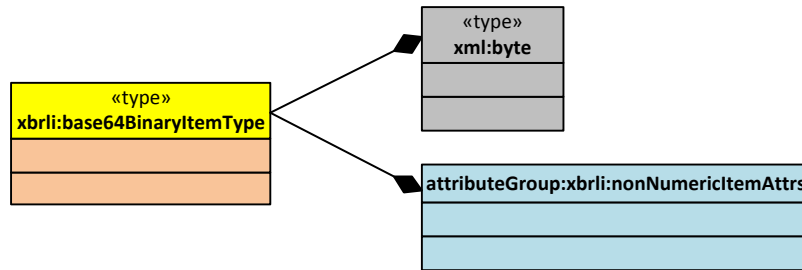
Tipo [no numérico](#) basado en el tipo XML “[NCName](#)”, extendido con el grupo de atributos correspondiente.

Tipo *xbrli:base64BinaryItemType*

1- Definición

```
<complexType name="base64BinaryItemType" final="extension">
  <simpleContent>
    <extension base="base64Binary">
      <attributeGroup ref="xbrli:nonNumericItemAttrs" />
    </extension>
  </simpleContent>
</complexType>
```

2- Modelo



3- Descripción

Tipo [no numérico](#) basado en el tipo XML “byte”, extendido con el grupo de atributos correspondiente.

Tipo *xbrli:booleanItemType*

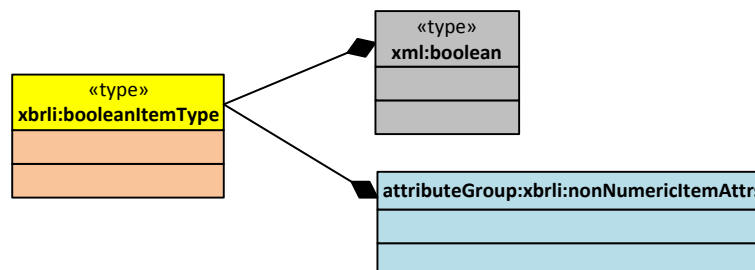
1- Definición

```

<complexType name="booleanItemType" final="extension">
  <simpleContent>
    <extension base="boolean">
      <attributeGroup ref="xbrli:nonNumericItemAttrs" />
    </extension>
  </simpleContent>
</complexType>

```

2- Modelo



3- Descripción

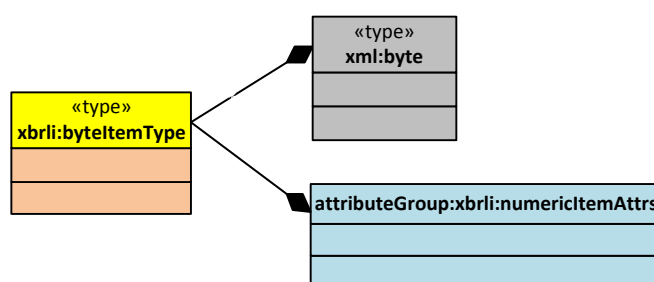
Tipo [no numérico](#) basado en el tipo XML “boolean”, extendido con el grupo de atributos correspondiente.

Tipo *xbrli:byteItemType*

1- Definición

```
<complexType name="byteItemType" final="extension">
  <simpleContent>
    <extension base="byte">
      <attributeGroup ref="xbrli:numericItemAttrs" />
    </extension>
  </simpleContent>
</complexType>
```

2- Modelo



3- Descripción

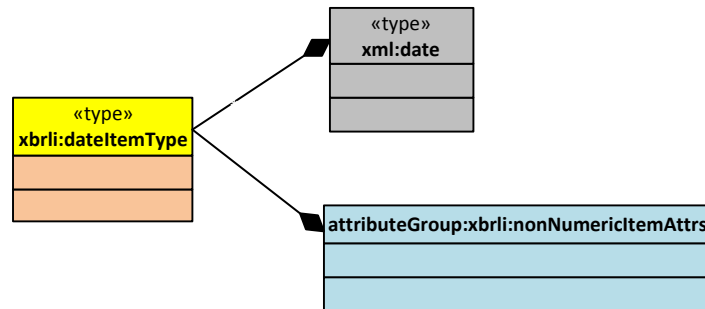
Tipo [numérico](#) basado en el tipo XML “byte”, extendido con el grupo de atributos correspondiente.

Tipo *xbrli:dateItemType*

1- Definición

```
<complexType name="dateItemType" final="extension">
  <simpleContent>
    <extension base="date">
      <attributeGroup ref="xbrli:nonNumericItemAttrs" />
    </extension>
  </simpleContent>
</complexType>
```

2- Modelo



3- Descripción

Tipo [no numérico](#) basado en el tipo XML “date” para expresar una fecha.

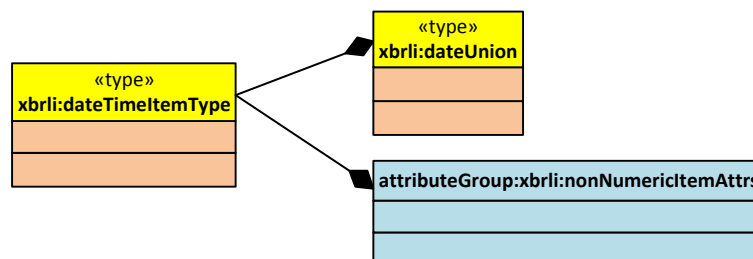
Tipo *xbrli:dateTimeItemType*

1- Definición

```

<complexType name="dateTimeItemType" final="extension">
  <simpleContent>
    <extension base="xbrli:dateUnion">
      <attributeGroup ref="xbrli:nonNumericItemAttrs" />
    </extension>
  </simpleContent>
</complexType>
  
```

2- Modelo



3- Descripción

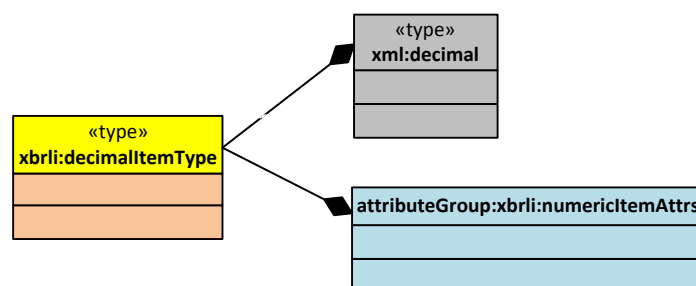
Tipo para expresar fecha y hora.

Tipo *xbrli:decimalItemType*

1- Definición

```
<complexType name="decimalItemType" final="extension">
  <simpleContent>
    <extension base="decimal">
      <attributeGroup ref="xbrli:numericItemAttrs" />
    </extension>
  </simpleContent>
</complexType>
```

2- Modelo



3- Descripción

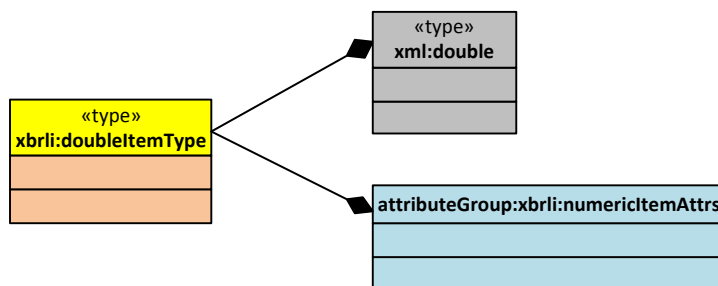
Tipo [numérico](#), basado en el tipo XML “decimal”, extendido con el grupo de atributos correspondiente.

Tipo *xbrli:doubleItemType*

1- Definición

```
<complexType name="doubleItemType" final="extension">
  <simpleContent>
    <extension base="double">
      <attributeGroup ref="xbrli:numericItemAttrs" />
    </extension>
  </simpleContent>
</complexType>
```

2- Modelo



3- Descripción

Tipo [numérico](#), basado en el tipo XML “double”, extendido con el grupo de atributos correspondiente.

Tipo *xbrli:durationItemType*

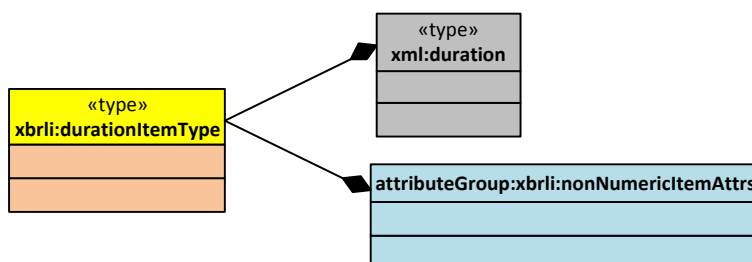
1- Definición

```

<complexType name="durationItemType" final="extension">
  <simpleContent>
    <extension base="duration">
      <attributeGroup ref="xbrli:nonNumericItemAttrs" />
    </extension>
  </simpleContent>
</complexType>

```

2- Modelo



3- Descripción

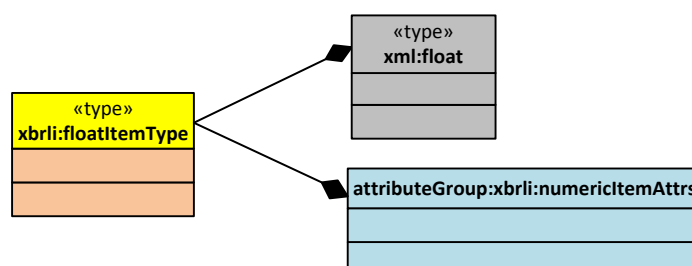
Tipo [no numérico](#), basado en el tipo XML “duration”, extendido con el grupo de atributos correspondiente.

Tipo xbrli:floatItemType

1- Definición

```
<complexType name="floatItemType" final="extension">
  <simpleContent>
    <extension base="float">
      <attributeGroup ref="xbrli:numericItemAttrs" />
    </extension>
  </simpleContent>
</complexType>
```

2- Modelo



3- Descripción

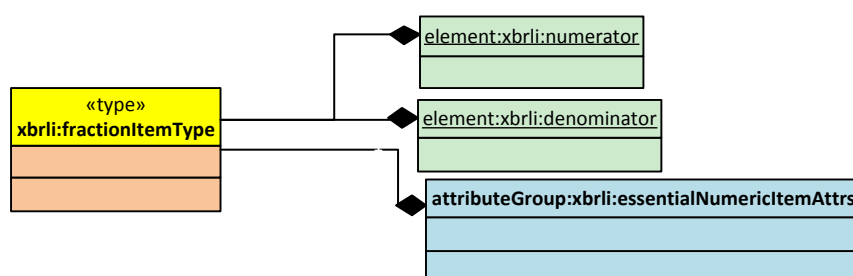
Tipo [numérico](#), basado en el tipo XML “float”, extendido con el grupo de atributos correspondiente.

Tipo xbrli:fractionItemType

1- Definición

```
<complexType name="fractionItemType" final="extension">
  <sequence>
    <element ref="xbrli:numerator" />
    <element ref="xbrli:denominator" />
  </sequence>
  <attributeGroup ref="xbrli:essentialNumericItemAttrs" />
</complexType>
```

2- Modelo



3- Descripción

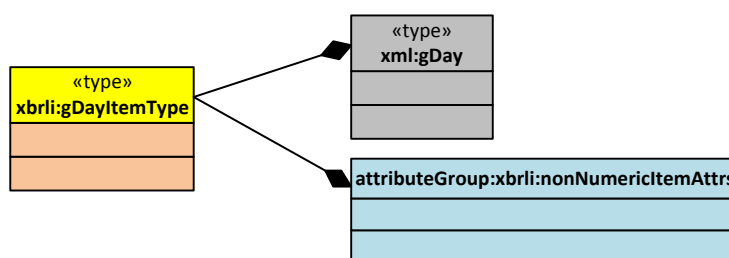
Tipo complejo que representa fracciones. Estará compuesto del elemento [numerator](#) (que está basado en el tipo “decimal”) y del elemento [denominator](#) (que no podrá ser 0, por ello, estará basado en el tipo “[nonZeroDecinmal](#)”).

Tipo xbrli:gDayItemType

1- Definición

```
<complexType name="gDayItemType" final="extension">
  <simpleContent>
    <extension base="gDay">
      <attributeGroup ref="xbrli:nonNumericItemAttrs" />
    </extension>
  </simpleContent>
</complexType>
```

2- Modelo



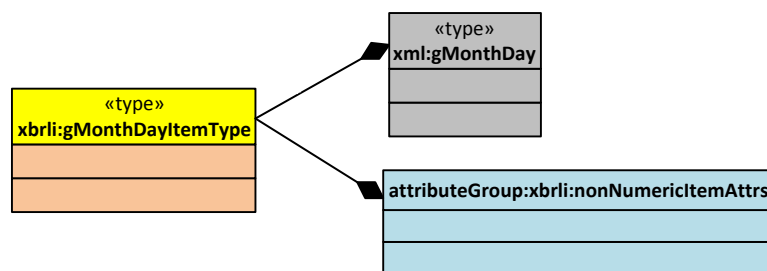
3- Descripción

Tipo [no numérico](#) basado en el tipo XML “gDay”, extendido con el grupo de atributos correspondiente.

1- Definición

```
<complexType name="gMonthDayItemType" final="extension">
  <simpleContent>
    <extension base="gMonthDay">
      <attributeGroup ref="xbrli:nonNumericItemAttrs" />
    </extension>
  </simpleContent>
</complexType>
```

2- Modelo



3- Descripción

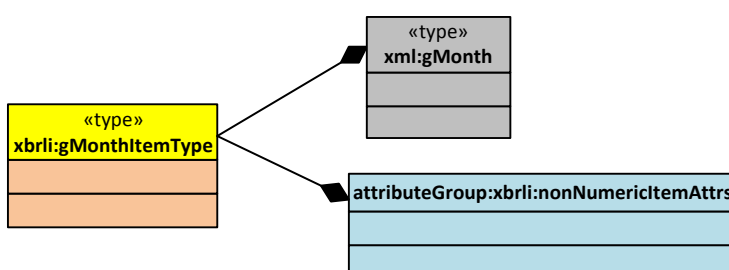
Tipo [no numérico](#) basado en el tipo XML “gMonthDay”, extendido con el grupo de atributos correspondiente.

Tipo xbrli:gMonthItemType

1- Definición

```
<complexType name="gMonthItemType" final="extension">
  <simpleContent>
    <extension base="gMonth">
      <attributeGroup ref="xbrli:nonNumericItemAttrs" />
    </extension>
  </simpleContent>
</complexType>
```

2- Modelo



3- Descripción

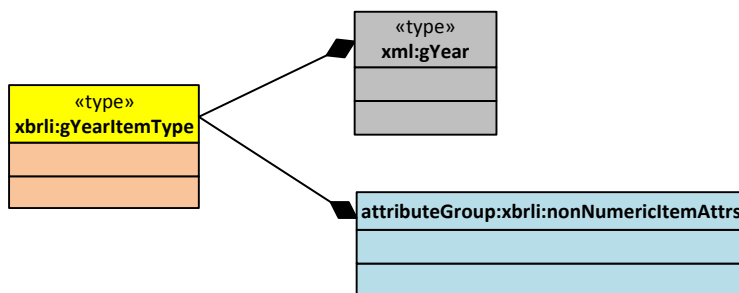
Tipo [no numérico](#) basado en el tipo XML “gMonth”, extendido con el grupo de atributos correspondiente.

Tipo xbrli:gYearItemType

1- Definición

```
<complexType name="gYearItemType" final="extension">
  <simpleContent>
    <extension base="gYear">
      <attributeGroup ref="xbrli:nonNumericItemAttrs" />
    </extension>
  </simpleContent>
</complexType>
```

2- Modelo



3- Descripción

Tipo [no numérico](#) basado en el tipo XML “gYear”, extendido con el grupo de atributos correspondiente.

Tipo xbrli:gYearMonthItemType

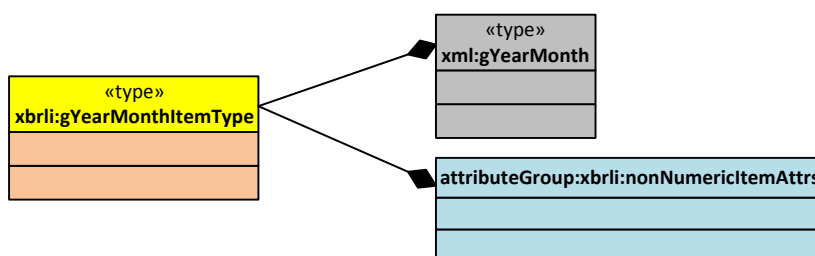
1- Definición

```

<complexType name="gYearMonthItemType" final="extension">
  <simpleContent>
    <extension base="gYearMonth">
      <attributeGroup ref="xbrli:nonNumericItemAttrs" />
    </extension>
  </simpleContent>
</complexType>

```

2- Modelo



3- Descripción

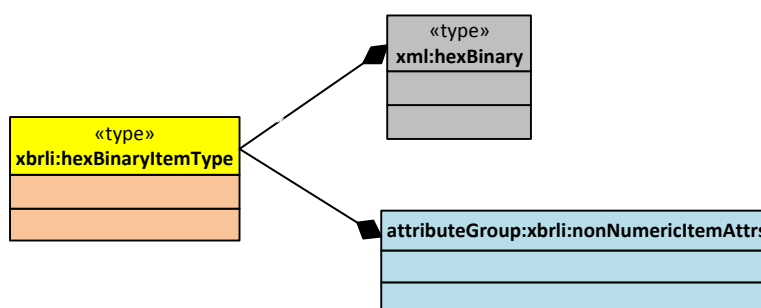
Tipo [no numérico](#) basado en el tipo XML “gYearMonthItemType”, extendido con el grupo de atributos correspondiente.

Tipo *xbrli:hexBinaryItemType*

1- Definición

```
<complexType name="hexBinaryItemType" final="extension">
  <simpleContent>
    <extension base="hexBinary">
      <attributeGroup ref="xbrli:nonNumericItemAttrs" />
    </extension>
  </simpleContent>
</complexType>
```

2- Modelo



3- Descripción

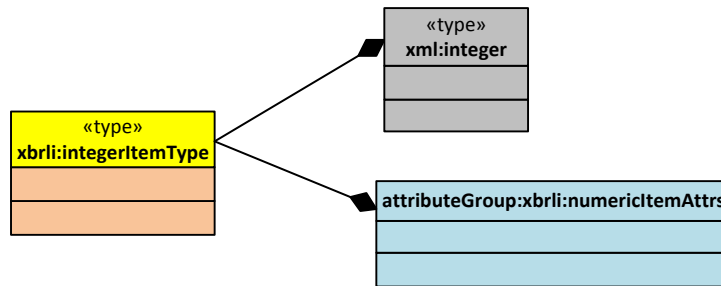
Tipo [no numérico](#) basado en el tipo XML “hexBinary”, extendido con el grupo de atributos correspondiente.

Tipo *xbrli:integerItemType*

1- Definición

```
<complexType name="integerItemType" final="extension">
  <simpleContent>
    <extension base="integer">
      <attributeGroup ref="xbrli:numericItemAttrs" />
    </extension>
  </simpleContent>
</complexType>
```

2- Modelo



3- Descripción

Tipo [numérico](#) basado en el tipo XML “integer”, extendido con el grupo de atributos correspondiente.

Tipo *xbrli:intItemType*

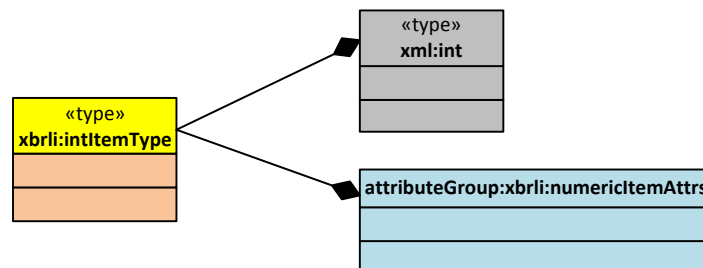
1- Definición

```

<complexType name="intItemType" final="extension">
  <simpleContent>
    <extension base="int">
      <attributeGroup ref="xbrli:numericItemAttrs" />
    </extension>
  </simpleContent>
</complexType>

```

2- Modelo



3- Descripción

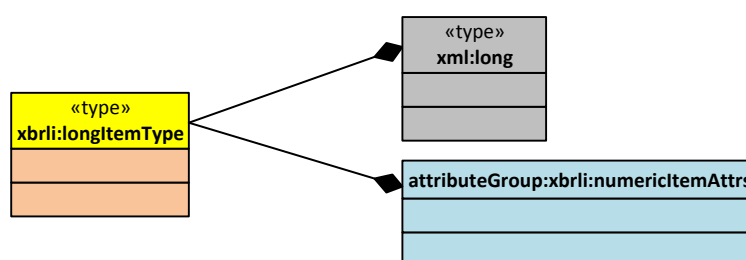
Tipo [numérico](#) basado en el tipo XML “int”, extendido con el grupo de atributos correspondiente.

Tipo *xbrli:longItemType*

1- Definición

```
<complexType name="longItemType" final="extension">
  <simpleContent>
    <extension base="long">
      <attributeGroup ref="xbrli:numericItemAttrs" />
    </extension>
  </simpleContent>
</complexType>
```

2- Modelo



3- Descripción

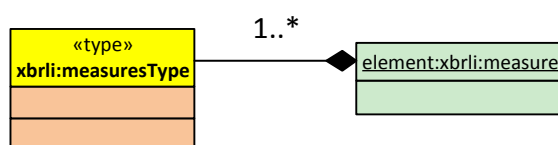
Tipo [numérico](#) basado en el tipo XML “long”, extendido con el grupo de atributos correspondiente.

Tipo *xbrli:measuresItemType*

1- Definición

```
<complexType name="measuresType">
  <annotation>
    <documentation>
      A collection of sibling measure elements
    </documentation>
  </annotation>
  <sequence>
    <element ref="xbrli:measure" minOccurs="1" maxOccurs="unbounded" />
  </sequence>
</complexType>
```

2- Modelo



3- Descripción

Tipo basado en una secuencia de elementos [measure](#). Indica el concepto que una unidad representa.

4- Ejemplo:

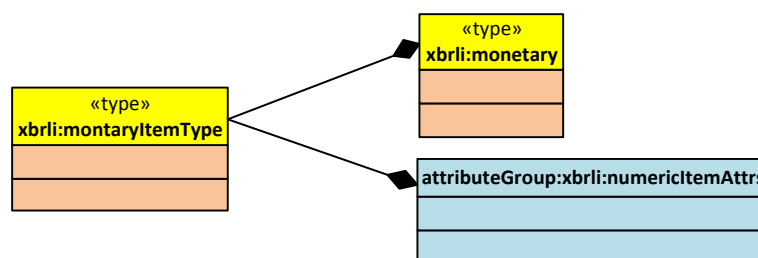
```
<unit id="u6">
  <divide>
    <unitNumerator>
      <measure>ISO4217:EUR</measure>
    </unitNumerator>
    <unitDenominator>
      <measure>xbrli:shares</measure>
    </unitDenominator>
  </divide>
</unit>
```

Tipo xbrli:monetaryItemType

1- Definición

```
<complexType name="monetaryItemType" final="extension">
  <simpleContent>
    <extension base="xbrli:monetary">
      <attributeGroup ref="xbrli:numericItemAttrs" />
    </extension>
  </simpleContent>
</complexType>
```

2- Modelo



3- Descripción

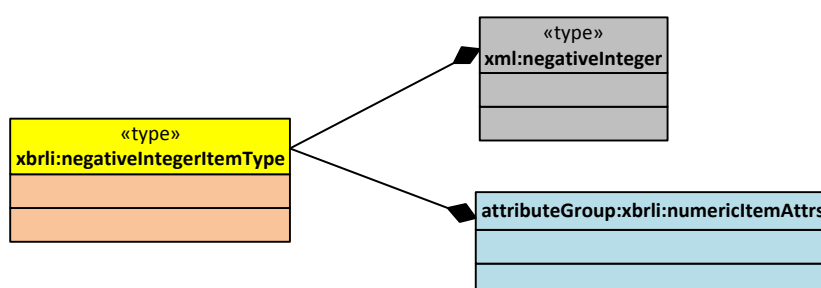
Tipo [numérico](#) basado en el tipo simple “[monetary](#)”, extendido con el grupo de atributos correspondiente.

Tipo *xbrli:negativeIntegerItemType*

1- Definición

```
<complexType name="negativeIntegerItemType" final="extension">
  <simpleContent>
    <extension base="negativeInteger">
      <attributeGroup ref="xbrli:numericItemAttrs" />
    </extension>
  </simpleContent>
</complexType>
```

2- Modelo



3- Descripción

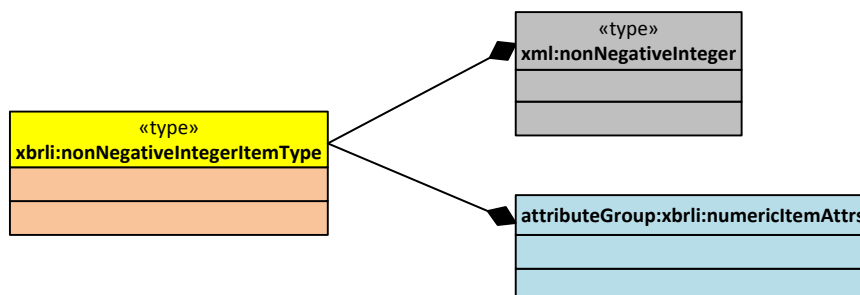
Tipo [numérico](#) basado en el tipo XML “negativeInteger”, extendido con el grupo de atributos correspondiente.

Tipo *xbrli:nonNegativeIntegerItemType*

1- Definición

```
<complexType name="nonNegativeIntegerItemType" final="extension">
  <simpleContent>
    <extension base="nonNegativeInteger">
      <attributeGroup ref="xbrli:numericItemAttrs" />
    </extension>
  </simpleContent>
</complexType>
```

2- Modelo



3- Descripción

Tipo [numérico](#) basado en el tipo XML “nonNegativeInteger”, extendido con el grupo de atributos correspondiente.

Tipo *xbrli:nonPositiveIntegerItemType*

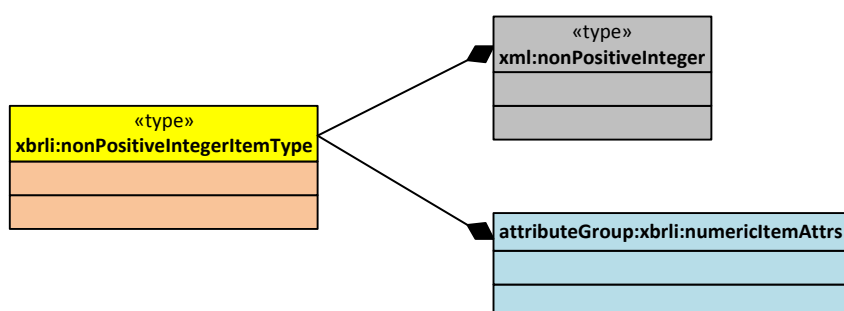
1- Definición

```

<complexType name="nonPositiveIntegerItemType" final="extension">
  <simpleContent>
    <extension base="nonPositiveInteger">
      <attributeGroup ref="xbrli:numericItemAttrs" />
    </extension>
  </simpleContent>
</complexType>

```

2- Modelo



3- Descripción

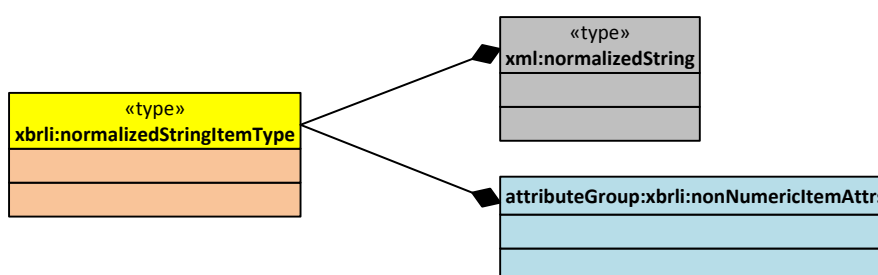
Tipo [numérico](#) basado en el tipo XML “nonPositiveInteger”, extendido con el grupo de atributos correspondiente.

Tipo *xbrli:normalizedStringItemType*

1- Definición

```
<complexType name="normalizedStringItemType" final="extension">
  <simpleContent>
    <extension base="normalizedString">
      <attributeGroup ref="xbrli:nonNumericItemAttrs" />
    </extension>
  </simpleContent>
</complexType>
```

2- Modelo



3- Descripción

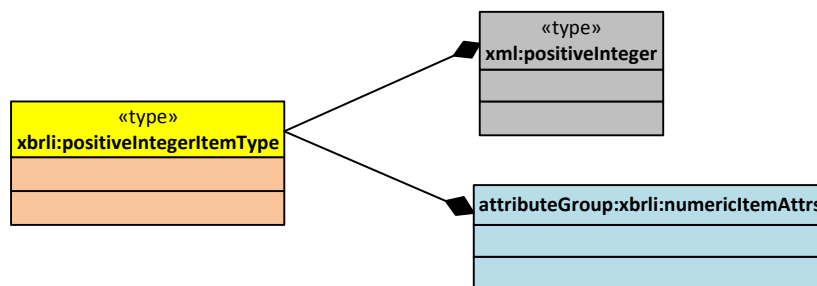
Tipo [no numérico](#) basado en el tipo XML “normalizedString”, extendido con el grupo de atributos correspondiente.

Tipo *xbrli:positiveIntegerItemType*

1- Definición

```
<complexType name="positiveIntegerItemType" final="extension">
  <simpleContent>
    <extension base="positiveInteger">
      <attributeGroup ref="xbrli:numericItemAttrs" />
    </extension>
  </simpleContent>
</complexType>
```

2- Modelo



3- Descripción

Tipo [numérico](#) basado en el tipo XML “PositiveInteger”, extendido con el grupo de atributos correspondiente.

Tipo xbrli:pureItemType

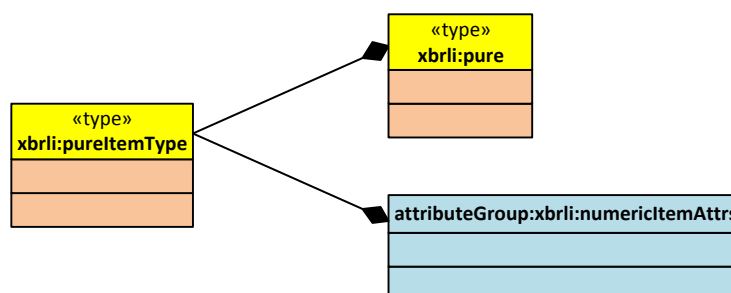
1- Definición

```

<complexType name="pureItemType" final="extension">
  <simpleContent>
    <extension base="xbrli:pure">
      <attributeGroup ref="xbrli:numericItemAttrs" />
    </extension>
  </simpleContent>
</complexType>

```

2- Modelo



3- Descripción

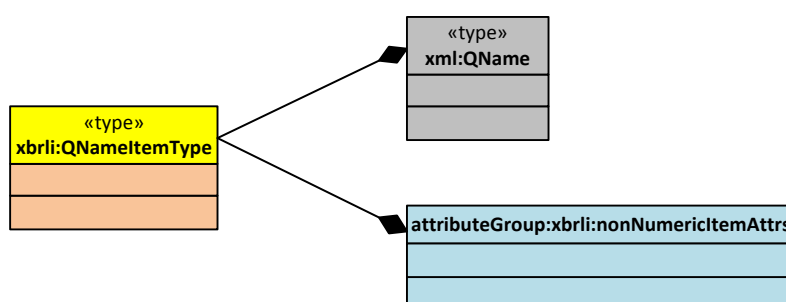
Tipo [numérico](#) basado en el tipo simple “[pure](#)”, y extendido por el grupo de atributos correspondiente.

Tipo *xbrli:QNameItem*Type

1- Definición

```
<complexType name="QNameItem" final="extension">
  <simpleContent>
    <extension base="QName">
      <attributeGroup ref="xbrli:nonNumericItemAttrs" />
    </extension>
  </simpleContent>
</complexType>
```

2- Modelo



3- Descripción

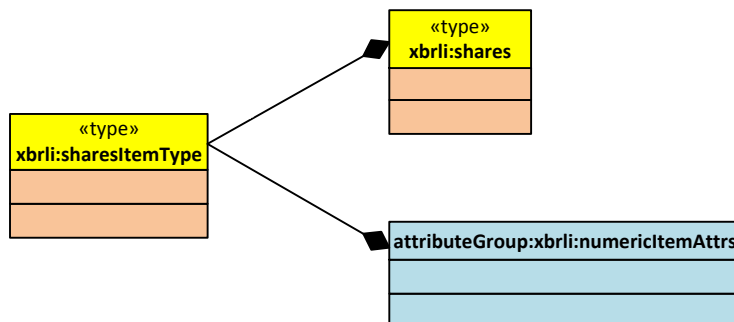
Tipo [no numérico](#) basado en el tipo XML “[QName](#)”, extendido con el grupo de atributos correspondiente.

Tipo *xbrli:sharesItem*Type

1- Definición

```
<complexType name="sharesItem" final="extension">
  <simpleContent>
    <extension base="xbrli:shares">
      <attributeGroup ref="xbrli:numericItemAttrs" />
    </extension>
  </simpleContent>
</complexType>
```

2- Modelo



3- Descripción

Tipo complejo [numérico](#) basado en el tipo simple “[shares](#)”, y extendido con el grupo de atributos correspondiente.

Tipo xbrli:shortItemType

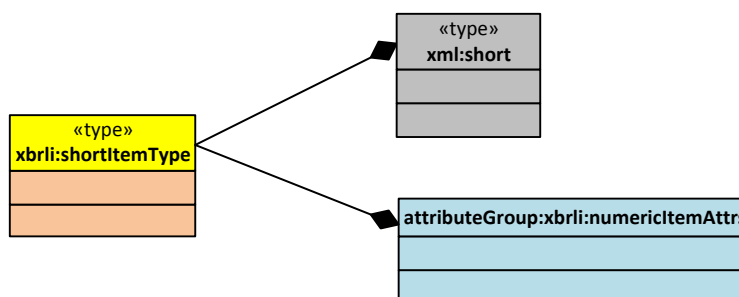
1- Definición

```

<complexType name="shortItemType" final="extension">
  <simpleContent>
    <extension base="short">
      <attributeGroup ref="xbrli:numericItemAttrs" />
    </extension>
  </simpleContent>
</complexType>

```

2- Modelo



3- Descripción

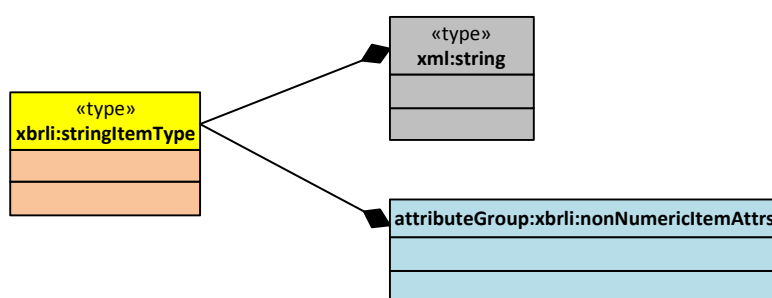
Tipo [numérico](#) basado en el tipo XML “short”, extendido con el grupo de atributos correspondiente.

Tipo *xbrli:stringItemType*

1- Definición

```
<complexType name="stringItemType" final="extension">
  <simpleContent>
    <extension base="string">
      <attributeGroup ref="xbrli:nonNumericItemAttrs" />
    </extension>
  </simpleContent>
</complexType>
```

2- Modelo



3- Descripción

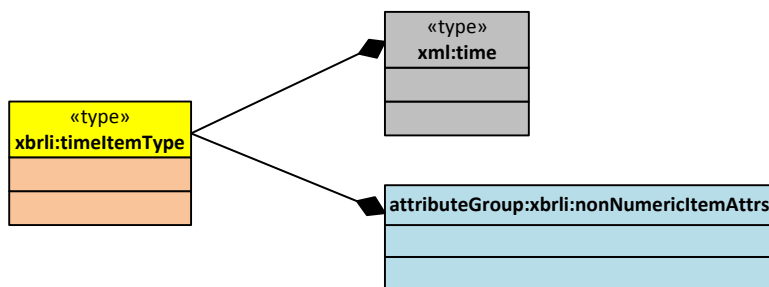
Tipo [no numérico](#) basado en el tipo XML “string”, extendido con el grupo de atributos correspondiente.

Tipo *xbrli:timeItemType*

1- Definición

```
<complexType name="timeItemType" final="extension">
  <simpleContent>
    <extension base="time">
      <attributeGroup ref="xbrli:nonNumericItemAttrs" />
    </extension>
  </simpleContent>
</complexType>
```

2- Modelo



3- Descripción

Tipo [no numérico](#) para expresar una hora, basado en el tipo XML “time”, extendido con el grupo de atributos correspondiente.

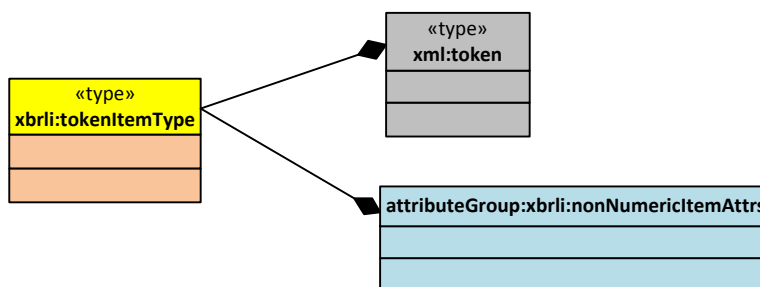
Tipo xbrli: tokenItemType

1- Definición

```

<complexType name="tokenItemType" final="extension">
  <simpleContent>
    <extension base="token">
      <attributeGroup ref="xbrli:nonNumericItemAttrs" />
    </extension>
  </simpleContent>
</complexType>
  
```

2- Modelo



3- Descripción

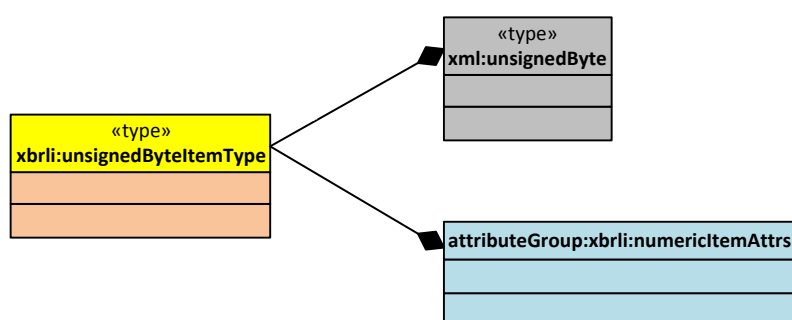
Tipo [no numérico](#) basado en el tipo XML “token”, extendido con el grupo de atributos correspondiente.

Tipo *xbrli:unsignedByteItem*Type

1- Definición

```
<complexType name="unsignedByteItem" final="extension">
  <simpleContent>
    <extension base="unsignedByte">
      <attributeGroup ref="xbrli:numericItemAttrs" />
    </extension>
  </simpleContent>
</complexType>
```

2- Modelo



3- Descripción

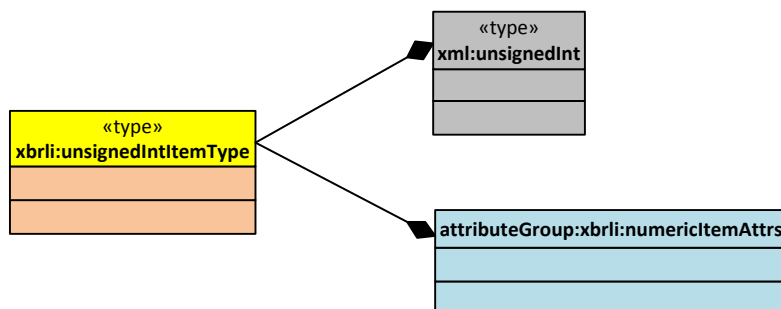
Tipo [numérico](#) sin signo de tipo byte, basado en el tipo XML “unsignedByte”, extendido con el grupo de atributos correspondiente.

Tipo *xbrli:unsignedIntItem*Type

1- Definición

```
<complexType name="unsignedIntItem" final="extension">
  <simpleContent>
    <extension base="unsignedInt">
      <attributeGroup ref="xbrli:numericItemAttrs" />
    </extension>
  </simpleContent>
</complexType>
```

2- Modelo



3- Descripción

Tipo [numérico](#) sin signo basado en el tipo XML “unsignedInt”, extendido con el grupo correspondiente.

Tipo *xbrli:unsignedLongItemType*

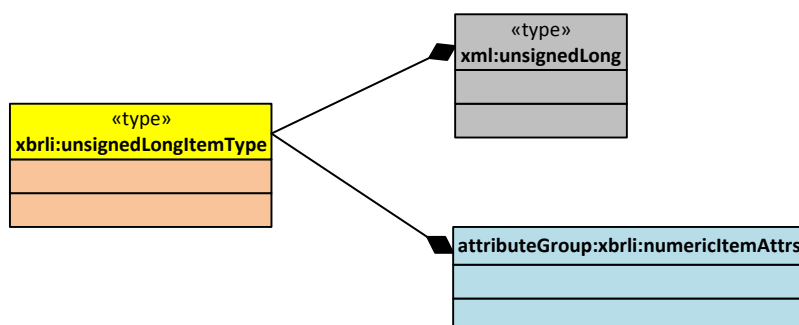
1- Definición

```

<complexType name="unsignedLongItemType" final="extension">
  <simpleContent>
    <extension base="unsignedLong">
      <attributeGroup ref="xbrli:numericItemAttrs" />
    </extension>
  </simpleContent>
</complexType>

```

2- Modelo



3- Descripción

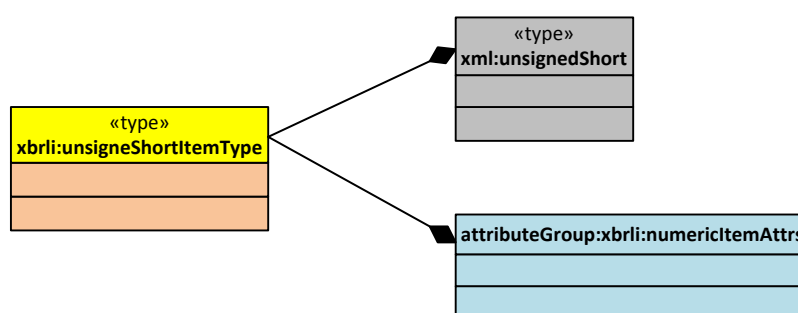
Tipo [numérico](#) sin signo basado en el tipo XML “unsignedLong”, extendido con el grupo de atributos correspondiente.

Tipo *xbrli:unsignedShortItemType*

1- Definición

```
<complexType name="unsignedShortItemType" final="extension">
  <simpleContent>
    <extension base="unsignedShort">
      <attributeGroup ref="xbrli:numericItemAttrs" />
    </extension>
  </simpleContent>
</complexType>
```

2- Modelo



3- Descripción

Tipo [numérico](#) sin signo basado en el tipo XML “unsignedShort”, extendido con el grupo de atributos correspondiente.

Tipo *xbrli:contextEntityType*

1- Definición

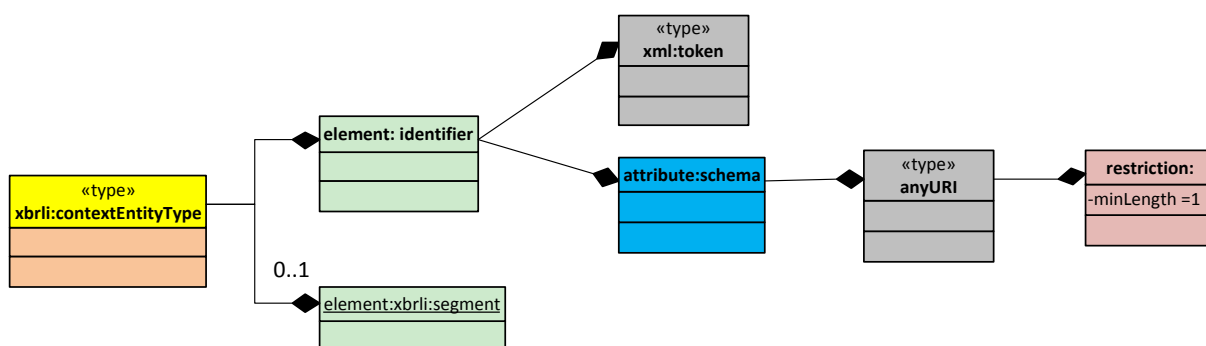
```
<complexType name="contextEntityType">
  <annotation>
    <documentation>
      The type for the entity element, used to describe the reporting entity.
      Note that the scheme attribute is required and cannot be empty.
    </documentation>
  </annotation>
  <sequence>
    <element name="identifier">
      <complexType>
        <simpleContent>
          <extension base="token">
            <attribute name="scheme" use="required">
              <simpleType>
```

```

    <restriction base="anyURI">
      <minLength value="1" />
    </restriction>
  </simpleType>
</attribute>
</extension>
</simpleContent>
</complexType>
</element>
<element ref="xbrli:segment" minOccurs="0" />
</sequence>
</complexType>

```

2- Modelo



3- Descripción

Asociado al elemento “[entity](#)”, indica el organismo o entidad que se asocia al hecho objeto del ítem. Está formado por el elemento “[identifier](#)” y el elemento “[segment](#)”, de forma optativa. Este último elemento definirá la dimensión del sector en el que se sitúa la entidad, y por tanto, será información valiosa que será conveniente informar.

4- Ejemplo [28]:

```

<xbrli:context id="Context_Instant_AjustesEliminaciones">
  <xbrli:entity>
    <xbrli:identifier scheme="http://www.ecb.int/stats/money/mfi">ES9000</xbrli:identifier>
    <xbrli:segment>
      <xbrldi:explicitMember dimension="es-be-cm-dim:Agrupacion">es-be-cm-
dim:AgrupacionIndividual</xbrldi:explicitMember>
    </xbrli:segment>
  </xbrli:entity>
</xbrli:period>
  <xbrli:instant>2008-09-30</xbrli:instant>
</xbrli:period>
<xbrli:scenario>

```

```

    <xbrldi:explicitMember dimension="es-be-d-FR-dist:DistribucionDimension">es-be-d-FR-
dist:AjustesEliminaciones</xbrldi:explicitMember>
  </xbrli:scenario>
</xbrli:context>

```

En el ejemplo podemos observar como existe una dimensión “entity”, con su atributo identificador “ES9000”, y la dimensión “segment” “es-be-cm-dim:Agrupacion”, con su atributo de dimensión “es-be-cm-dim:AgrupacionIndividual”.

Tipo xbrli:contextPeriodType

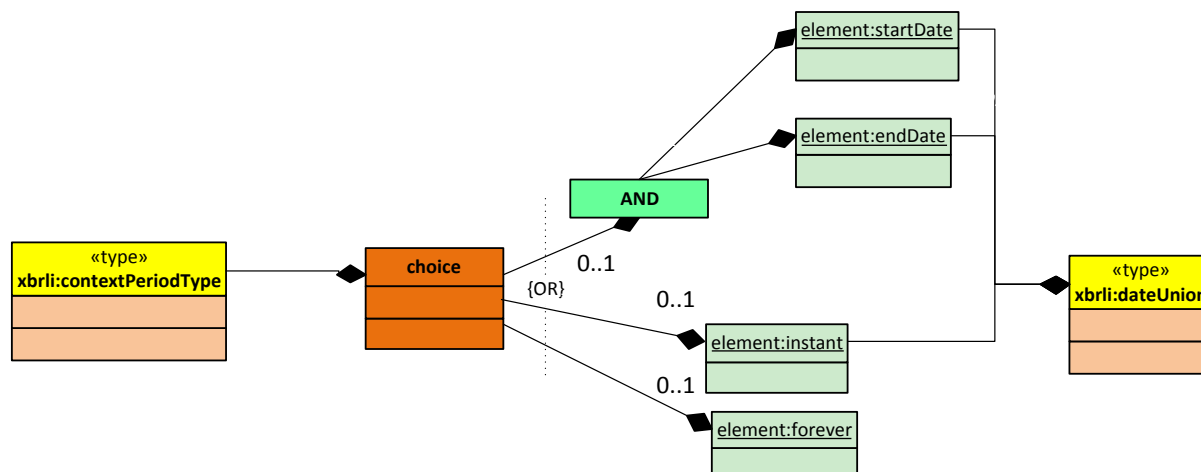
1- Definición

```

<complexType name="contextPeriodType">
  <annotation>
    <documentation>
      The type for the period element, used to describe the reporting date info.
    </documentation>
  </annotation>
  <choice>
    <sequence>
      <element name="startDate" type="xbrli:dateUnion" />
      <element name="endDate" type="xbrli:dateUnion" />
    </sequence>
    <element name="instant" type="xbrli:dateUnion" />
    <element name="forever">
      <complexType />
    </element>
  </choice>
</complexType>

```

2- Modelo



3- Descripción

Tipo de datos para expresar el instante o intervalo de tiempo que referencia a un elemento. Puede estar representado bien por un intervalo de tiempo, con fecha de inicio y fin (“startDate”, “endDate”), bien por un punto determinado en el tiempo, o bien por el elemento “forever”.

4- Ejemplo [27]:

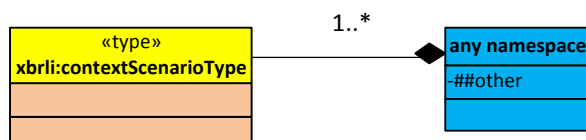
```
<xbrli:context id="e_x7_x20_x14_eq0d_x11_x1">
  <xbrli:entity>
    <xbrli:identifier scheme="http://scheme">abc</xbrli:identifier>
  </xbrli:entity>
  <xbrli:period>
    <xbrli:instant>2011-06-12</xbrli:instant>
  </xbrli:period>
  <xbrli:scenario>
    <xbrldi:explicitMember dimension="dim:AS">dCT:x7</xbrldi:explicitMember>
    <xbrldi:explicitMember dimension="dim:AT">dAT:x20</xbrldi:explicitMember>
    <xbrldi:explicitMember dimension="dim:CS">dSE:x14</xbrldi:explicitMember>
    <xbrldi:explicitMember dimension="dim:DL">dTI:eq0d</xbrldi:explicitMember>
    <xbrldi:explicitMember dimension="dim:PL">dPL:x11</xbrldi:explicitMember>
    <xbrldi:explicitMember dimension="dim:RS">drs:x1</xbrldi:explicitMember>
  </xbrli:scenario>
</xbrli:context>
```

En este caso, tenemos que el identificador de la entidad es “abc”. El periodo, un instante, “2011-06-12” (una fecha concreta), y el escenario, el conjunto de dimensiones “AS”, “AT”, “CS”, “DL”, “PL” y “RS”, con sus respectivos atributos definidos, que serían “dCT:x7”, “dAT:x20”, “dSE:x14”, “dTI:eq0d”, “dPL:x11” y “drs:x1”. En otras ocasiones nos encontraremos con periodos formados por intervalos de tiempo, con fecha de inicio y fecha final, y en otras con el elemento “forever”, que define que es para siempre.

1- Definición

```
<complexType name="contextScenarioType">
  <annotation>
    <documentation>
      Used for the scenario under which fact have been reported.
    </documentation>
  </annotation>
  <sequence>
    <any namespace="##other" processContents="lax"
      minOccurs="1" maxOccurs="unbounded" />
  </sequence>
</complexType>
```

2- Modelo



3- Descripción

Asociado al elemento “scenario”, describe información adicional que debe ser conocida en el propio informe. Estará formado por una secuencia de espacios de nombres.

4- Ejemplo [28]:

```
<xbrli:context id="Context_Instant_OtrasEntidades">
  <xbrli:entity>
    <xbrli:identifier scheme="http://www.ecb.int/stats/money/mfi">ES9000</xbrli:identifier>
    <xbrli:segment>
      <xbrldi:explicitMember dimension="es-be-cm-dim:Agrupacion">es-be-cm-
dim:AgrupacionIndividual</xbrldi:explicitMember>
    </xbrli:segment>
  </xbrli:entity>
  <xbrli:period>
    <xbrli:instant>2008-09-30</xbrli:instant>
  </xbrli:period>
  <xbrli:scenario>
    <xbrldi:explicitMember dimension="es-be-d-FR-dist:DistribucionDimension">es-be-d-FR-
dist:OtrasEntidades</xbrldi:explicitMember>
  </xbrli:scenario>
</xbrli:context>
```

En este ejemplo (y otros similares que veremos a lo largo del proyecto) se puede observar que un escenario estará formado por un espacio de nombres, que definirán las dimensiones que forman el escenario, y por tanto el contexto.

4.3.5 Elementos XBRL

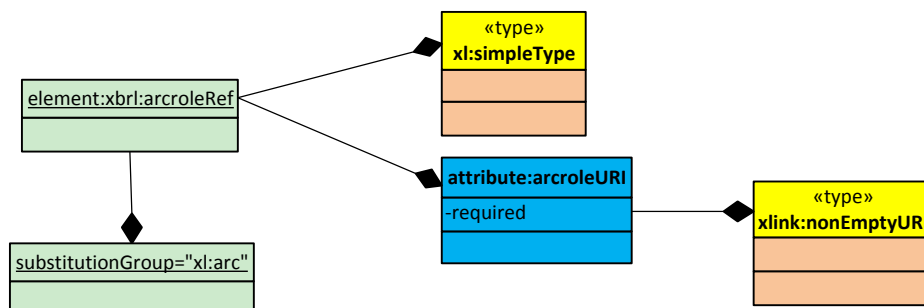
A continuación detallaremos los elementos que forman parte del meta-metamodelo de XBRL.

Elemento *link:arcroleRef*

1- Definición

```
<element name="arcroleRef" substitutionGroup="xl:simple">
  <annotation>
    <documentation>
      Definition of the roleRef element - used
      to link to resolve xlink:arcrole attribute values to
      the arcroleType element declaration.
    </documentation>
  </annotation>
  <complexType>
    <complexContent>
      <extension base="xl:simpleType">
        <attribute name="arcroleURI" type="xlink:nonEmptyURI" use="required">
          <annotation>
            <documentation>
              This attribute contains the arc role name.
            </documentation>
          </annotation>
        </attribute>
      </extension>
    </complexContent>
  </complexType>
</element>
```

2- Modelo



3- Descripción

Elemento que referencia la definición de los valores de cualquier atributo [arcrole](#) usados en los elementos enlace “[footnote](#)” y que aparecen inmediatamente después de los “[roleRef](#)”.

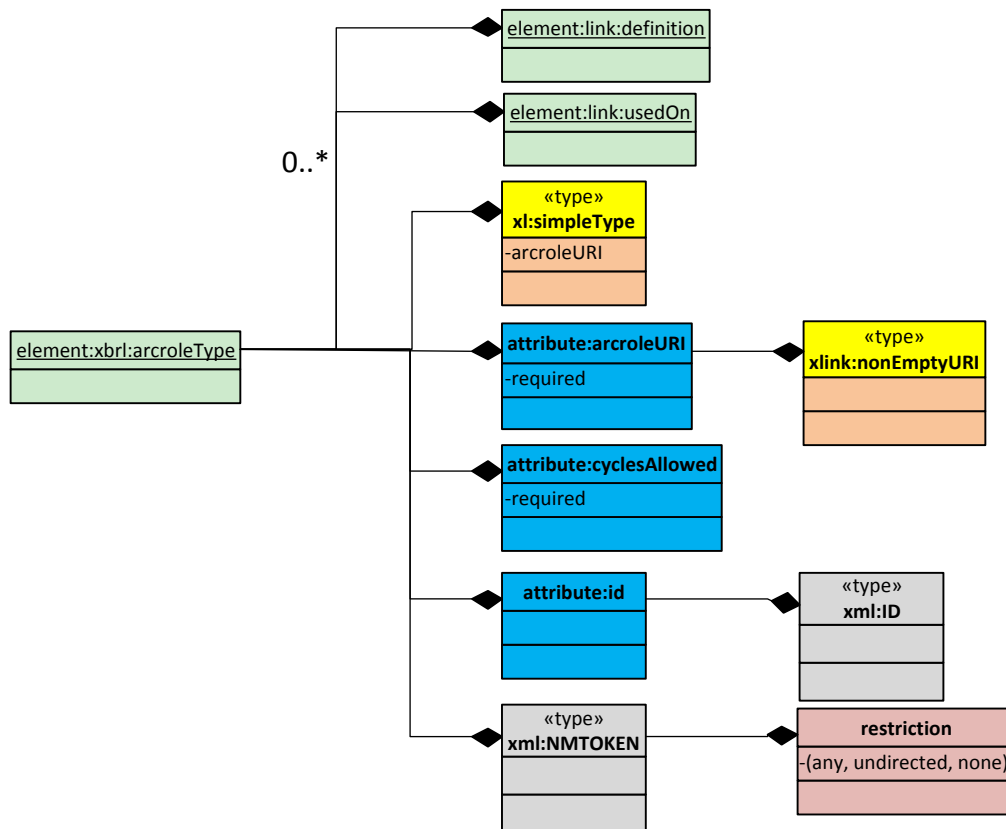
Es una extensión del tipo simple "[simpleType](#)", que añade un atributo del tipo "[nonEmptyURI](#)". Sirven, entre otras cosas, para referenciar las "linkbases".

Elemento link:arcroleType

1- Definición

```
<element name="arcroleType">
  <annotation>
    <documentation>
      The arcroleType element definition - used to define custom
      arc role values in XBRL extended links.
    </documentation>
  </annotation>
  <complexType>
    <sequence>
      <element ref="link:definition" minOccurs="0"/>
      <element ref="link:usedOn" maxOccurs="unbounded"/>
    </sequence>
    <attribute name="arcroleURI" type="xlink:nonEmptyURI" use="required"/>
    <attribute name="id" type="ID"/>
    <attribute name="cyclesAllowed" use="required">
      <simpleType>
        <restriction base="NMTOKEN">
          <enumeration value="any"/>
          <enumeration value="undirected"/>
          <enumeration value="none"/>
        </restriction>
      </simpleType>
    </attribute>
  </complexType>
</element>
```

2- Modelo



3- Descripción

Elemento que especifica los valores del atributo “[arcrole](#)” en un enlace personalizado de una aplicación.

Está compuesto de una secuencia de elementos “[definition](#)” y “[usedOn](#)”, un tipo “[simpleType](#)”, tres atributos, “arcroleURI”, “cyclesAllowed” e “id”, todos obligatorios y un tipo simple basado XML “NMTOKEN”.

4- Ejemplo [26]:

```

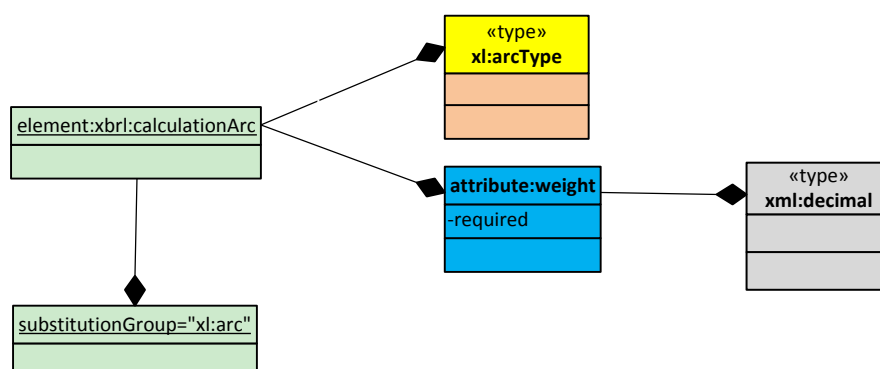
<link:arcroleType arcroleURI="http://www.example.com/" cyclesAllowed="any">
  <link:definition>string</link:definition>
  <link:usedOn>QName</link:usedOn>
</link:arcroleType>

```

1- Definición

```
<element name="calculationArc" substitutionGroup="xl:arc">
  <complexType>
    <annotation>
      <documentation>
        Extension of the extended link arc type for calculation arcs.
        Adds a weight attribute to track weights on contributions to
        summations.
      </documentation>
    </annotation>
    <complexContent>
      <extension base="xl:arcType">
        <attribute name="weight" type="decimal" use="required"/>
      </extension>
    </complexContent>
  </complexType>
</element>
```

2- Modelo



3- Descripción

Tipo basado en el tipo `arcType`, extendido por el atributo `weight`, cuyo valor es un decimal positivo o negativo. Define cómo los conceptos se relacionan entre sí para determinados cálculos

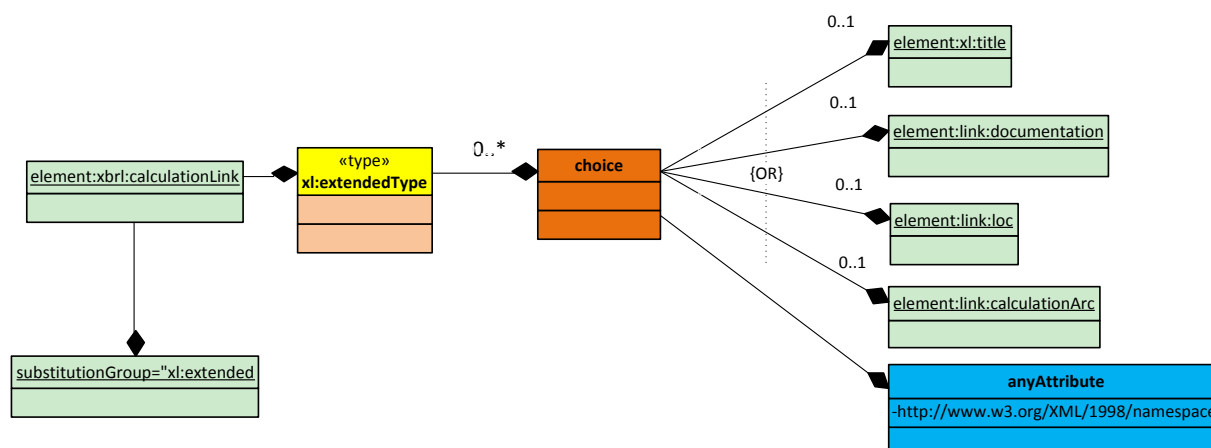
4- Ejemplo [27]:

```
<calculationArc
  xlink:type="arc"
  xlink:arcrole="http://www.xbrl.org/2003/arcrole/summation-item"
  xlink:from="AssetsTotal" xlink:to="AssetsCurrent"
  weight="1.0" order="1"/>
```

1- Definición

```
<element name="calculationLink" substitutionGroup="xl:extended">
  <annotation>
    <documentation>
      calculation extended link element definition
    </documentation>
  </annotation>
  <complexType>
    <complexContent>
      <restriction base="xl:extendedType">
        <choice minOccurs="0" maxOccurs="unbounded">
          <element ref="xl:title"/>
          <element ref="link:documentation"/>
          <element ref="link:loc"/>
          <element ref="link:calculationArc"/>
        </choice>
      </restriction>
    </complexContent>
  </complexType>
</element>
```

2- Modelo



3- Descripción

Elemento que define una o varias formas de calcular unos ítems respecto de otros. Contiene relaciones matemáticas tales como sumas y restas ponderadas entre ellos. Las aplicaciones utilizarán estos elementos para validar los valores de las instancias, y evaluar su consistencia. Por ejemplo, puede servir en el cálculo de totales.

4- Ejemplo [8]:

```

<calculationLink xlink:type="extended" xlink:role="http://www.w3.org/2003/role/link">
  <loc xlink:type="locator" xlink:href="esquema.xsd#ifrs-gp_TotalInmovilizado"
  xlink:label="A"/>
  <loc xlink:type="locator" xlink:href="esquema.xsd#ifrs-gp_GastosEstablecimiento"
  xlink:label="B"/>
  <calculationArc xlink:type="arc" xlink:from="A" xlink:to="B"
  xlink:arcrole="http://www.w3.org/2003/arcrole/summation-item"/>
  order="1" weight="1.0" use="optional"/>
</calculationLink>

```

Elemento link:definition

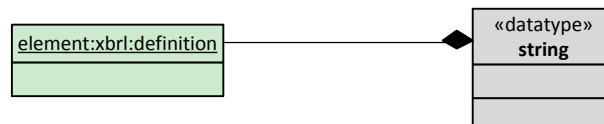
1- Definición

```

<element name="definition" type="string">
  <annotation>
    <documentation>
      The element to use for human-readable definition
      of custom roles and arc roles.
    </documentation>
  </annotation>
</element>

```

2- Modelo



3- Descripción

Elemento que proporciona una definición legible y entendible referida a los atributos [“role”](#) y [“arcrole”](#).

4- Ejemplo:

```

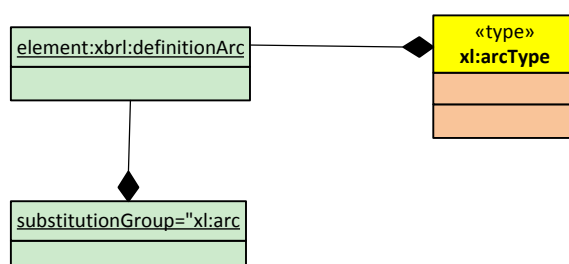
<link:roleType roleURI="http://www.eurofiling.info/dsi/ct_11" id="dsict_11">
  <link:definition>Debt securities held by original currency and counterparty residence (Part
  2)</link:definition>
  <link:usedOn>link:definitionLink</link:usedOn>
</link:roleType>

```

1- Definición

```
<element name="definitionArc" type="xl:arcType" substitutionGroup="xl:arc">
  <annotation>
    <documentation>
      Concrete arc for use in definition extended links.
    </documentation>
  </annotation>
</element>
```

2- Modelo



3- Descripción

Elemento consistente en un [arco](http://www.w3.org/2003/arcrole/arcrole) que define los distintos tipos de relaciones que pueden aparecer entre los conceptos. Puede tomar cuatro valores:

- "general-special" - ("<http://www.w3.org/2003/arcrole/general-special>")
- "essence-alias" - ("<http://www.w3.org/2003/arcrole/essence-alias>")
- "similar-tuples" - ("<http://www.w3.org/2003/arcrole/similar-tuples>")
- "requires-element" - ("<http://www.w3.org/2003/arcrole/requires-element>")

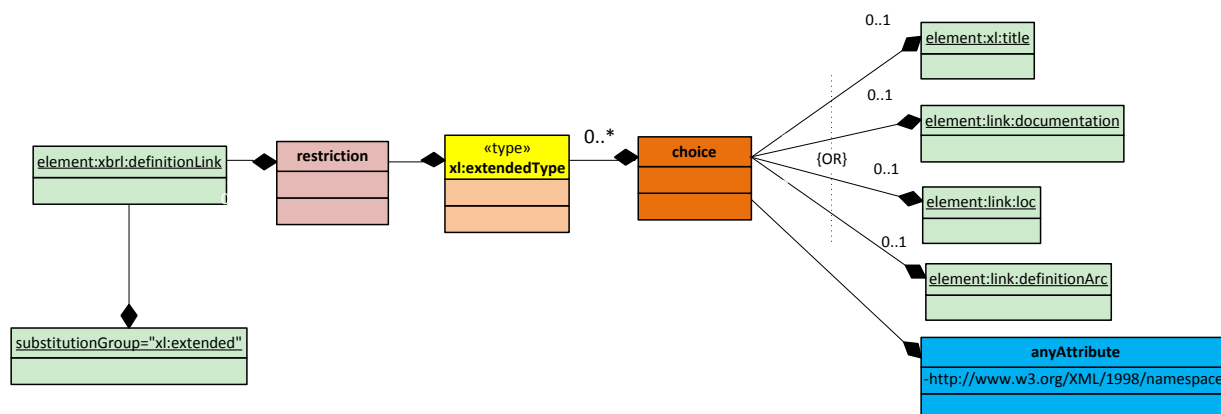
4- Ejemplo [27]:

```
<link:definitionArc xlink:type="arc"
  xlink:arcrole="http://xbrl.org/int/dim/arcrole/hypercube-dimension"
  xlink:from="PatrimonioNetoPublicoConsolidadoHypercube"
  xlink:to="PatrimonioNetoDimension"
  xlink:title="definición: PatrimonioNetoPublicoConsolidadoHypercube to
PatrimonioNetoDimension"
  order="1.0"/>
```

1- Definición

```
<element name="definitionLink" substitutionGroup="xl:extended">
  <annotation>
    <documentation>
      definition extended link element definition
    </documentation>
  </annotation>
  <complexType>
    <complexContent>
      <restriction base="xl:extendedType">
        <choice minOccurs="0" maxOccurs="unbounded">
          <element ref="xl:title"/>
          <element ref="link:documentation"/>
          <element ref="link:loc"/>
          <element ref="link:definitionArc"/>
        </choice>
        <anyAttribute namespace="http://www.w3.org/XML/1998/namespace"
processContents="lax" />
      </restriction>
    </complexContent>
  </complexType>
</element>
```

2- Modelo



3- Descripción

Elemento que sirve para representar relaciones entre conceptos y hechos.

4- Ejemplo: [27]

```
<link:definitionLink xlink:type="extended"
```

```

    xlink:role="http://www.bde.es/es/fr/esrs/finrep/6-2008/2008-11-
26/role/EstadoFlujosEfectivoPublicoConsolidado">
    <link:loc xlink:type="locator"
        xlink:href="es-be-t-FINREP-III4-
EstadoTotalCambiosPatrimonioNetoPublicoConsolidado.xsd#es-be-t-FR-
PNPC_PatrimonioNetoPublicoConsolidadoHypercube"
        xlink:label="PatrimonioNetoPublicoConsolidadoHypercube"
        xlink:title="PatrimonioNetoPublicoConsolidadoHypercube"/>
    <link:loc xlink:type="locator"
        xlink:href="es-be-d-FINREP-EstadoTotalCambiosPatrimonioNeto.xsd#es-be-d-FR-
pn_PatrimonioNetoDimension"
        xlink:label="PatrimonioNetoDimension"
        xlink:title="PatrimonioNetoDimension"/>
    <link:definitionArc
        xlink:type="arc"
        xlink:arcrole="http://xbrl.org/int/dim/arcrole/hypercube-dimension"
        xlink:from="PatrimonioNetoPublicoConsolidadoHypercube" xlink:to="PatrimonioNetoDimension"
        xlink:title="definition: PatrimonioNetoPublicoConsolidadoHypercube to
PatrimonioNetoDimension" order="1.0"/>
    <link:loc xlink:type="locator"
        xlink:href="es-be-d-FINREP-EstadoTotalCambiosPatrimonioNeto.xsd#es-be-d-FR-
pn_TotalPatrimonioNeto"
        xlink:label="TotalPatrimonioNeto"
        xlink:title="TotalPatrimonioNeto"/>
    <link:definitionArc xlink:type="arc"
        xlink:arcrole="http://xbrl.org/int/dim/arcrole/dimension-domain"
        xlink:from="PatrimonioNetoDimension"
        xlink:to="TotalPatrimonioNeto"
        xlink:title="definition: PatrimonioNetoDimension to TotalPatrimonioNeto"
        order="1.0"/>
</link:definitionLink>

```

Elemento link:documentation

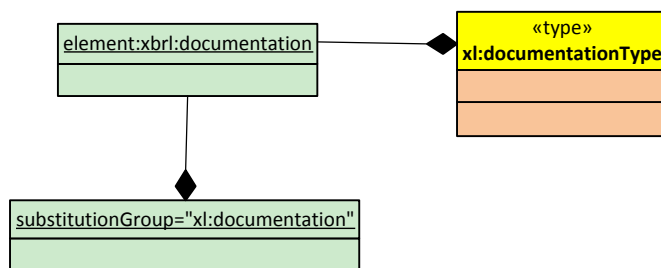
1- Definición

```

<element name="documentation" type="xl:documentationType"
    substitutionGroup="xl:documentation">
    <annotation>
        <documentation>
            Concrete element to use for documentation of
            extended links and linkbases.
        </documentation>
    </annotation>
</element>

```

2- Modelo



3- Descripción

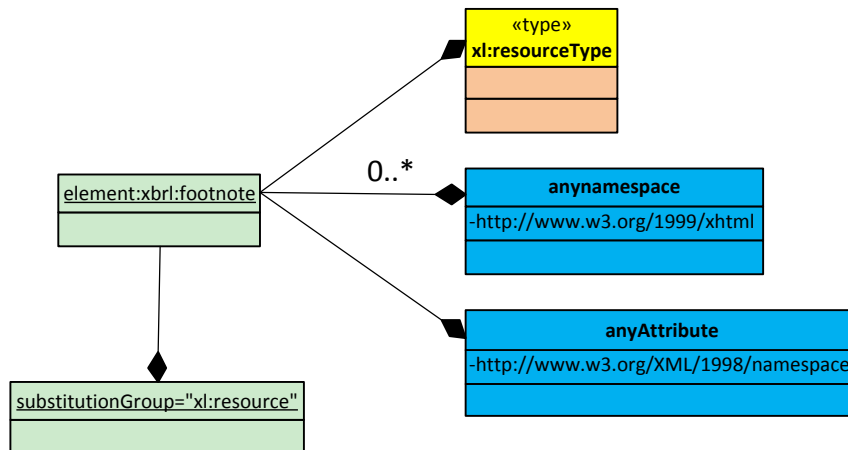
Elemento utilizado para documentar los vínculos o enlaces extendidos y linkbases.

Elemento *link:footnote*

1- Definición

```
<element name="footnote" substitutionGroup="xl:resource">
  <annotation>
    <documentation>
      Definition of the reference resource element
    </documentation>
  </annotation>
  <complexType mixed="true">
    <complexContent mixed="true">
      <extension base="xl:resourceType">
        <sequence>
          <any namespace="http://www.w3.org/1999/xhtml" processContents="skip" minOccurs="0"
maxOccurs="unbounded"/>
        </sequence>
          <anyAttribute namespace="http://www.w3.org/XML/1998/namespace"
processContents="lax"/>
        </extension>
      </complexContent>
    </complexType>
  </element>
```

2- Modelo



3- Descripción

Elemento que es el único permitido como recurso en los elementos “[footnoteLink](#)”, restringiendo su contenido a recursos genéricos. Concretamente, tiene contenido mixto: una cadena, un fragmento de XHTML, o una combinación de ambos [17].

Permiten describir detalladamente los hechos económicos de una instancia XBRL.

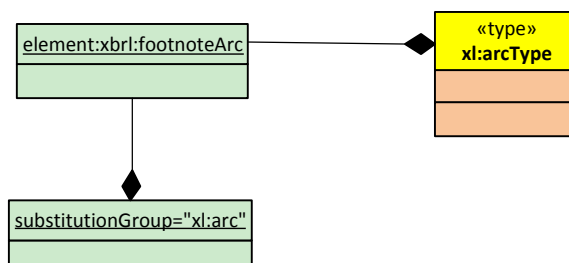
Elemento [link:footnoteArc](#)

1- Definición

```

<element name="footnoteArc" type="xl:arcType" substitutionGroup="xl:arc">
  <annotation>
    <documentation>
      Concrete arc for use in footnote extended links.
    </documentation>
  </annotation>
</element>
  
```

2- Modelo



3- Descripción

Elemento que es un arco. Aparece en un documento instancia en un vínculo extendido “[footnoteLink](#)”, que conecta hechos con información adicional legible

4- Ejemplo [27]:

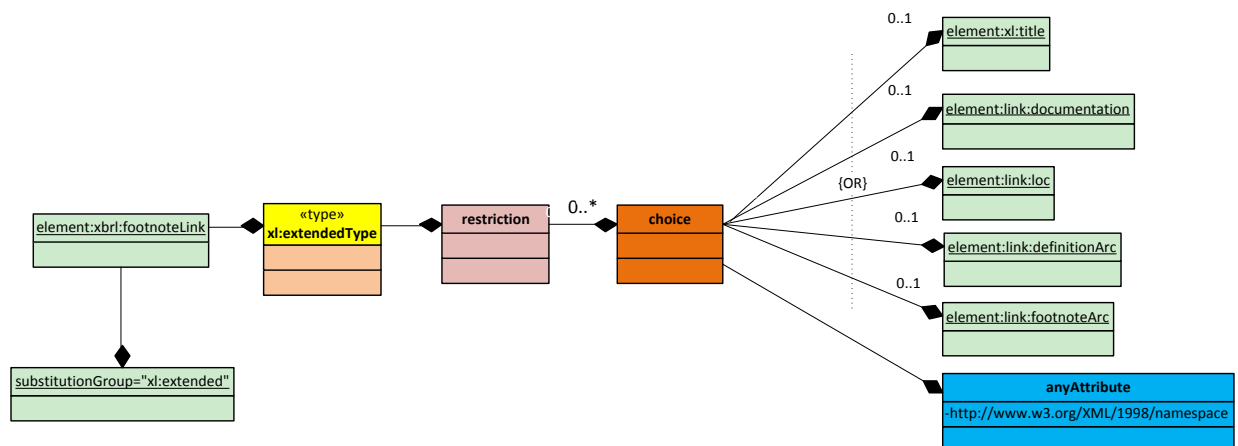
```
<link:footnoteArc  
  xlink:type="arc"  
  xlink:arcrole="http://www.xbrl.org/2003/arcrole/fact-footnote"  
  xlink:from="ifrs:gp_ContributionsReceivedInAdvance"  
  xlink:to="ifrs:gp_ContributionsNote" order="1.0"/>
```

Elemento link:footnoteLink

1- Definición

```
<element name="footnoteLink" substitutionGroup="xl:extended">  
  <annotation>  
    <documentation>  
      footnote extended link element definition  
    </documentation>  
  </annotation>  
  <complexType>  
    <complexContent>  
      <restriction base="xl:extendedType">  
        <choice minOccurs="0" maxOccurs="unbounded">  
          <element ref="xl:title"/>  
          <element ref="link:documentation"/>  
          <element ref="link:loc"/>  
          <element ref="link:footnoteArc"/>  
          <element ref="link:footnote"/>  
        </choice>  
        <anyAttribute namespace="http://www.w3.org/XML/1998/namespace"  
processContents="lax" />  
      </restriction>  
    </complexContent>  
  </complexType>  
</element>
```

2- Modelo



3- Descripción

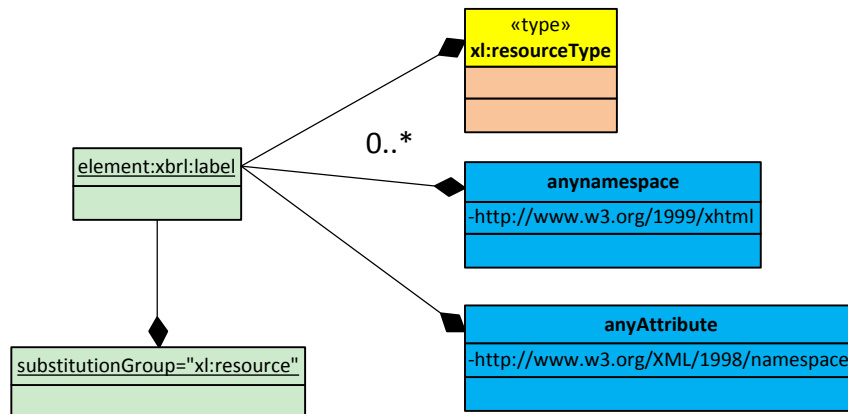
Elemento que define la relación de notas a pie de página con su documento instancia. Puesto que no son los hechos los que tienen esta información, si no el propio documento instancia, es necesaria la definición de un elemento de este tipo.

Elemento *link:label*

1- Definición

```
<element name="label" substitutionGroup="xl:resource">
  <annotation>
    <documentation>
      Definition of the label resource element.
    </documentation>
  </annotation>
  <complexType mixed="true">
    <complexContent mixed="true">
      <extension base="xl:resourceType">
        <sequence>
          <any namespace="http://www.w3.org/1999/xhtml" processContents="skip" minOccurs="0"
maxOccurs="unbounded"/>
        </sequence>
        <anyAttribute namespace="http://www.w3.org/XML/1998/namespace"
processContents="lax"/>
      </extension>
    </complexContent>
  </complexType>
</element>
```

2- Modelo



3- Descripción

Documenta conceptos XBRL, siendo por lo tanto un recurso.

4- Ejemplo [23]:

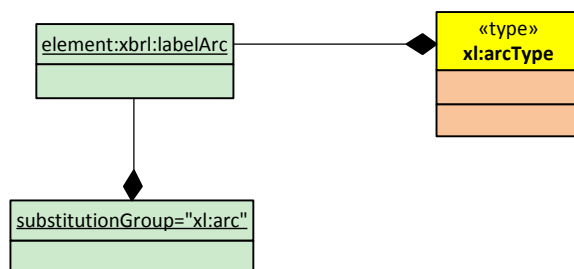
```
<labelLink xlink:type="extended"
  xlink:role="http://www.xbrl.org/2003/role/link">
  <loc xlink:type="locator" xlink:label="a" xlink:href="example.xsd#conceptA"/>
  <label xlink:type="resource" xml:lang="en" xlink:label="lab_a"
xlink:role="http://www.xbrl.org/2003/role/label">Concept A</label>
  <label xlink:type="resource" xml:lang="en" xlink:label="lab_a"
xlink:role="http://www.xbrl.org/2003/role/totalLabel">Total of Concept A</label>
  <labelArc xlink:type="arc" xlink:from="a" xlink:to="lab_a"
  xlink:arcrole="http://www.xbrl.org/2003/arcrole/concept-label"/>
</labelLink>
```

Elemento link:labelArc

1- Definición

```
<element name="labelArc" type="xl:arcType" substitutionGroup="xl:arc">
  <annotation>
    <documentation>
      Concrete arc for use in label extended links.
    </documentation>
  </annotation>
</element>
```

2- Modelo



3- Descripción

Elemento encargado de crear la relación entre un elemento inicial y otro final, sin restricciones, creando una asociación entre concepto y recurso.

4- Ejemplo [23]:

```

<labelLink xlink:type="extended"
  xlink:role="http://www.xbrl.org/2003/role/link">
  <loc xlink:type="locator" xlink:label="a" xlink:href="example.xsd#conceptA"/>
  <label xlink:type="resource" xml:lang="en" xlink:label="lab_a"
xlink:role="http://www.xbrl.org/2003/role/label">Concept A</label>
  <label xlink:type="resource" xml:lang="en" xlink:label="lab_a"
xlink:role="http://www.xbrl.org/2003/role/totalLabel">Total of Concept A</label>
  <labelArc xlink:type="arc" xlink:from="a" xlink:to="lab_a"
  xlink:arcrole="http://www.xbrl.org/2003/arcrole/concept-label"/>
</labelLink>
  
```

Elemento link:labelLink

1- Definición

```

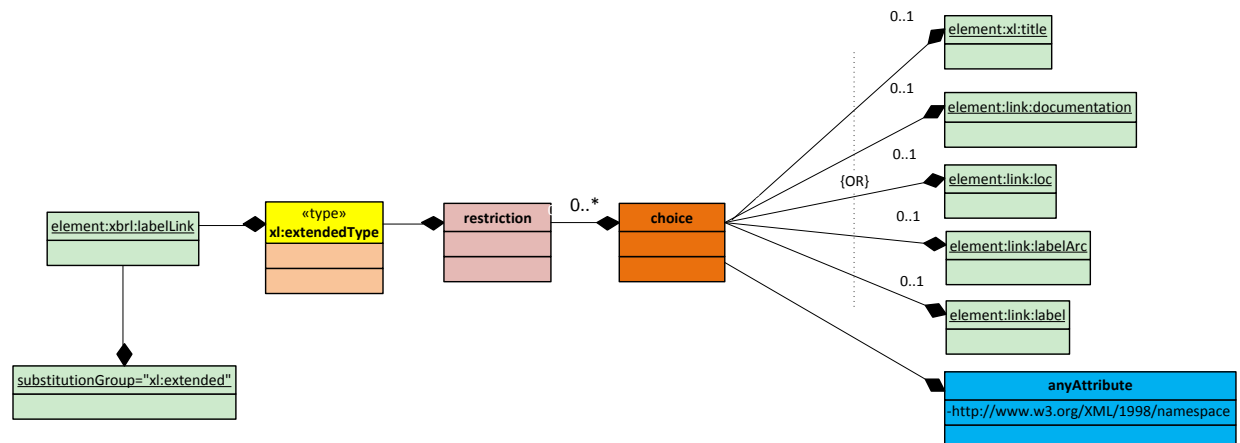
<element name="labelLink" substitutionGroup="xl:extended">
  <annotation>
    <documentation>
      label extended link element definition
    </documentation>
  </annotation>
  <complexType>
    <complexContent>
      <restriction base="xl:extendedType">
        <choice minOccurs="0" maxOccurs="unbounded">
          <element ref="xl:title"/>
          <element ref="link:documentation"/>
          <element ref="link:loc"/>
          <element ref="link:labelArc"/>
          <element ref="link:label"/>
        </choice>
      </restriction>
    </complexContent>
  </complexType>
</element>
  
```

```

    </choice>
    <anyAttribute namespace="http://www.w3.org/XML/1998/namespace"
processContents="lax" />
  </restriction>
</complexContent>
</complexType>
</element>

```

2- Modelo



3- Descripción

Elemento que relaciona concepto con información local a la linkbase, esto es, representada en su mismo documento.

Elemento link:linkbase

1- Definición

```

<element name="linkbase">
  <annotation>
    <documentation>
      Definition of the linkbase element. Used to
      contain a set of zero or more extended link elements.
    </documentation>
  </annotation>
  <complexType>
    <choice minOccurs="0" maxOccurs="unbounded">
      <element ref="link:documentation"/>
      <element ref="link:roleRef"/>
      <element ref="link:arcroleRef"/>
      <element ref="xl:extended"/>
    </choice>
    <attribute name="id" type="ID" use="optional"/>
  </complexType>
</element>

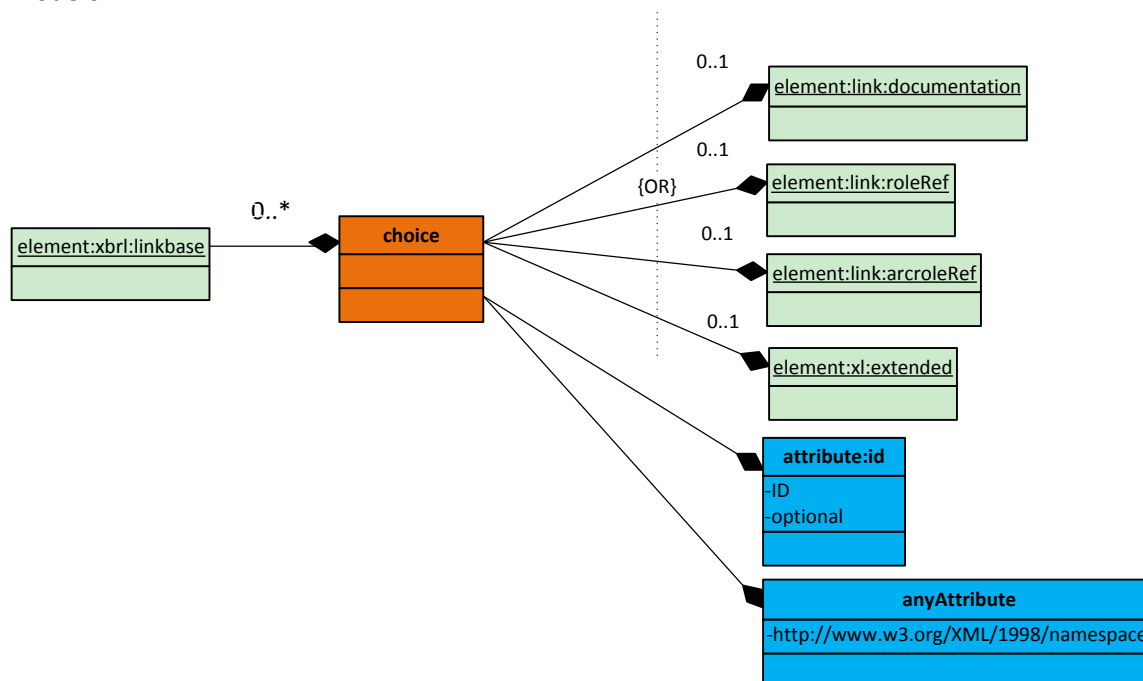
```

```

    <anyAttribute namespace="http://www.w3.org/XML/1998/namespace" processContents="lax"/>
  </complexType>
</element>

```

2- Modelo



3- Descripción

Completa las restricciones que los esquemas normativos de XBRL imponen sobre la especificación XLink, para poder manejar los enlaces que este lenguaje necesita.

Elemento *link:linkbaseRef*

1- Definición

```

<element name="linkbaseRef" substitutionGroup="xl:simple">
  <annotation>
    <documentation>
      Definition of the linkbaseRef element - used
      to link to XBRL taxonomy extended links from
      taxonomy schema documents and from XBRL
      instances.
    </documentation>
  </annotation>
  <complexType>
    <complexContent>
      <restriction base="xl:simpleType">
        <attribute ref="xlink:arcrole" use="required">
          <annotation>

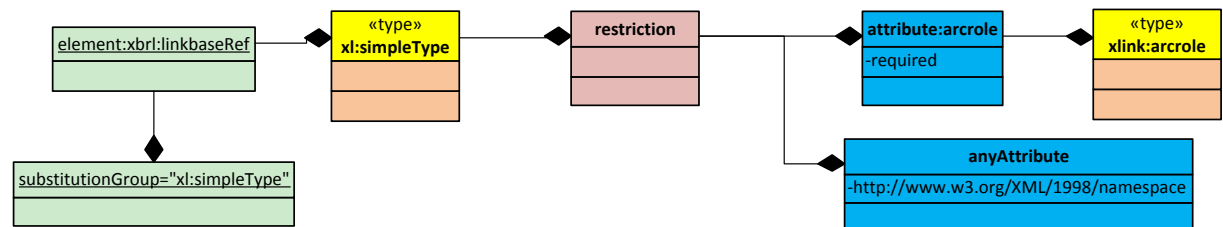
```

```

<documentation>
This attribute must have the value:
http://www.w3.org/1999/xlink/properties/linkbase
</documentation>
</annotation>
</attribute>
<anyAttribute namespace="http://www.w3.org/XML/1998/namespace"
processContents="lax" />
</restriction>
</complexContent>
</complexType>
</element>

```

2- Modelo



3- Descripción

Elemento que sirve para que los documentos instancia referencien a los documentos linkbase que forman parte del DTS que le da soporte.

4- Ejemplo [23]:

```

<link:linkbaseRef
xlink:type="simple"
xlink:href="linkbase_reference.xml"
xlink:role="http://www.xbrl.org/2003/role/referenceLinkbaseRef"
xlink:arcrole="http://www.w3.org/1999/xlink/properties/linkbase"/>

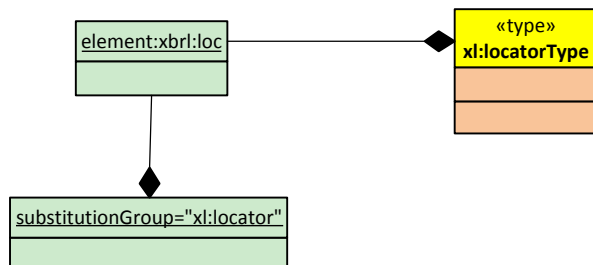
```


Elemento link:loc

1- Definición

```
<element name="loc" type="xl:locatorType" substitutionGroup="xl:locator">
  <annotation>
    <documentation>
      Concrete locator element. The loc element is the
      XLink locator element for all extended links in XBRL.
    </documentation>
  </annotation>
</element>
```

2- Modelo



3- Descripción

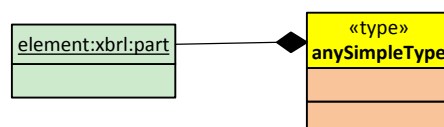
Elemento que actua de localizador para cualquier enlace XBRL.

Elemento link:part

1- Definición

```
<element name="part" type="anySimpleType" abstract="true">
  <annotation>
    <documentation>
      Definition of the reference part element - for use in reference resources.
    </documentation>
  </annotation>
</element>
```

2- Modelo



3- Descripción

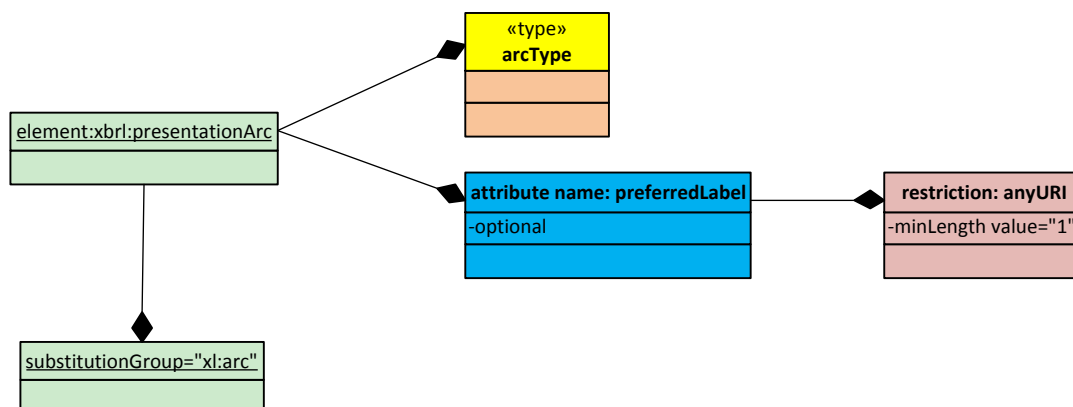
Elemento abstracto definido en la especificación XBRL que puede ser usado en las taxonomías como [grupo de sustitución](#) para los elementos definidos en las mismas, permitiéndoles ser incluidos en elementos referencia.

Elemento *link:presentationArc*

1- Definición

```
<element name="presentationArc" substitutionGroup="xl:arc">
  <complexType>
    <annotation>
      <documentation>
        Extension of the extended link arc type for presentation arcs.
        Adds a preferredLabel attribute that documents the role attribute
        value of preferred labels (as they occur in label extended links).
      </documentation>
    </annotation>
    <complexContent>
      <extension base="xl:arcType">
        <attribute name="preferredLabel" use="optional">
          <simpleType>
            <restriction base="anyURI">
              <minLength value="1"/>
            </restriction>
          </simpleType>
        </attribute>
      </extension>
    </complexContent>
  </complexType>
</element>
```

2- Modelo



3- Descripción

Elemento de tipo arco, utilizado en linkbases de presentación para definir las relaciones jerárquicas entre los conceptos en términos de su aparición en los informes de negocios.

4- Ejemplo [27]:

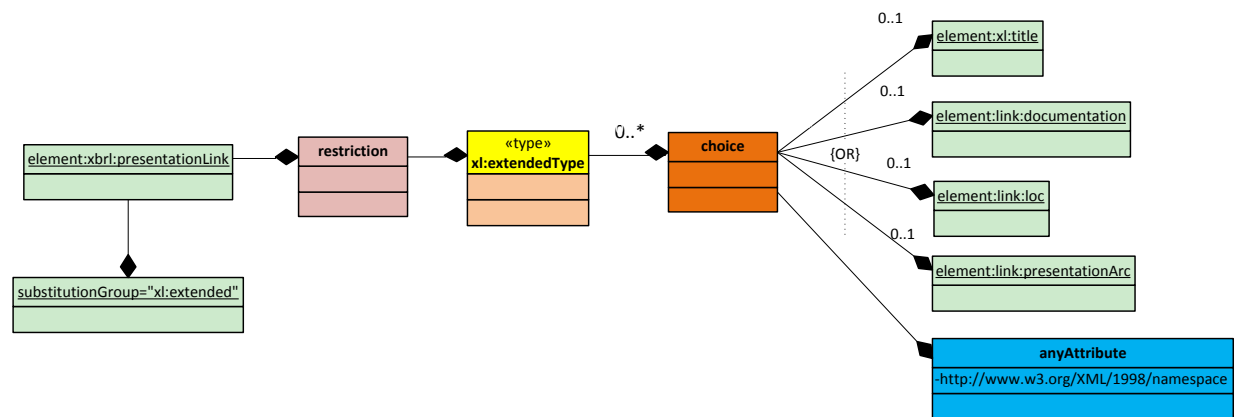
```
<presentationArc
xlink:type="arc"
xlink:arcrole="http://www.xbrl.org/2003/arcrole/parent-child"
xlink:from="ifrs-gp_Assets"
xlink:to="ifrs-gp_AssetsCurrent"
order="1" use="optional"/>
```

Elemento link:presentationLink

1- Definición

```
<element name="presentationLink" substitutionGroup="xl:extended">
  <annotation>
    <documentation>
      presentation extended link element definition.
    </documentation>
  </annotation>
  <complexType>
    <complexContent>
      <restriction base="xl:extendedType">
        <choice minOccurs="0" maxOccurs="unbounded">
          <element ref="xl:title"/>
          <element ref="link:documentation"/>
          <element ref="link:loc"/>
          <element ref="link:presentationArc"/>
        </choice>
        <anyAttribute namespace="http://www.w3.org/XML/1998/namespace"
processContents="lax" />
      </restriction>
    </complexContent>
  </complexType>
</element>
```

2- Modelo



3- Descripción

Elemento que almacena la linkbase de presentación, que tiene información sobre las relaciones entre los elementos.

Elemento link:reference

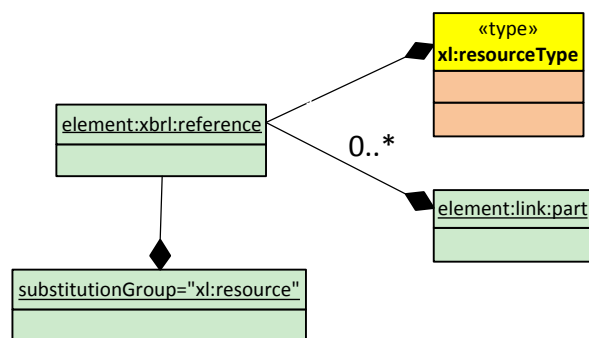
1- Definición

```

<element name="reference" substitutionGroup="xl:resource">
  <annotation>
    <documentation>
      Definition of the reference resource element.
    </documentation>
  </annotation>
  <complexType mixed="true">
    <complexContent mixed="true">
      <extension base="xl:resourceType">
        <sequence>
          <element ref="link:part" minOccurs="0" maxOccurs="unbounded"/>
        </sequence>
      </extension>
    </complexContent>
  </complexType>
</element>

```

2- Modelo



3- Descripción

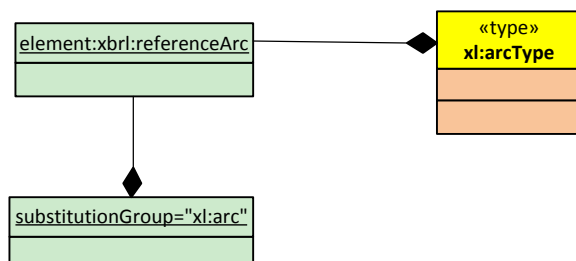
Elemento que permite que las taxonomías XBRL puedan basar sus declaraciones conceptuales en fuentes externas y autorizadas.

Elemento *link:referenceArc*

1- Definición

```
<element name="referenceArc" type="xl:arcType" substitutionGroup="xl:arc">
  <annotation>
    <documentation>
      Concrete arc for use in reference extended links.
    </documentation>
  </annotation>
</element>
```

2- Modelo



3- Descripción

Elemento arco que enlaza conceptos con recursos “[reference](#)”, sin ninguna restricción.

Elemento *link:referenceLink*

1- Definición

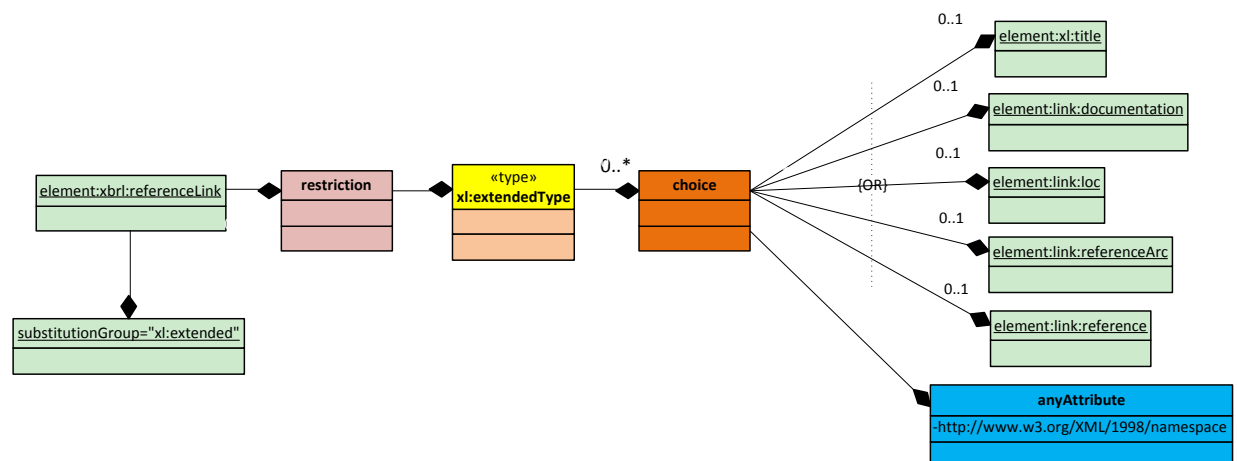
```
<element name="referenceLink" substitutionGroup="xl:extended">
```

```

<annotation>
  <documentation>
    reference extended link element definition
  </documentation>
</annotation>
<complexType>
  <complexContent>
    <restriction base="xl:extendedType">
      <choice minOccurs="0" maxOccurs="unbounded">
        <element ref="xl:title"/>
        <element ref="link:documentation"/>
        <element ref="link:loc"/>
        <element ref="link:referenceArc"/>
        <element ref="link:reference"/>
      </choice>
      <anyAttribute namespace="http://www.w3.org/XML/1998/namespace"
processContents="lax" />
    </restriction>
  </complexContent>
</complexType>
</element>

```

2- Modelo



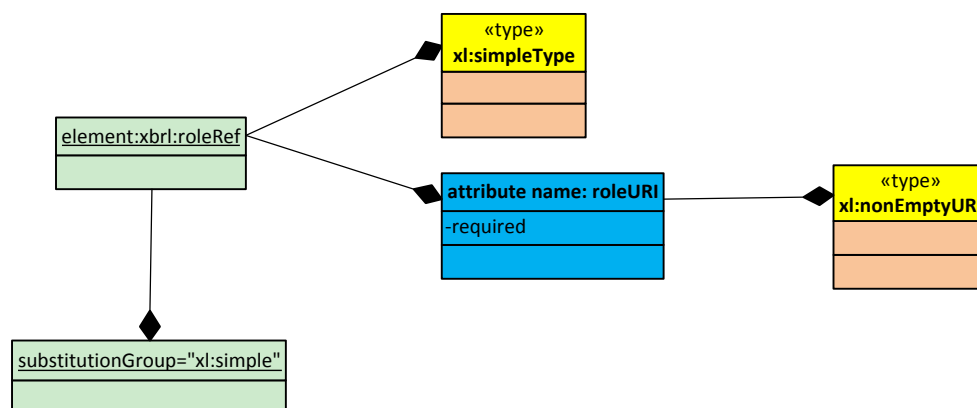
3- Descripción

Elemento que representa el enlace a fuentes autorizadas para dar significado a los conceptos usados.

1- Definición

```
<element name="roleRef" substitutionGroup="xl:simple">
  <annotation>
    <documentation>
      Definition of the roleRef element - used
      to link to resolve xlink:role attribute values to
      the roleType element declaration.
    </documentation>
  </annotation>
  <complexType>
    <complexContent>
      <extension base="xl:simpleType">
        <attribute name="roleURI" type="xlink:nonEmptyURI" use="required">
          <annotation>
            <documentation>
              This attribute contains the role name.
            </documentation>
          </annotation>
        </attribute>
      </extension>
    </complexContent>
  </complexType>
</element>
```

2- Modelo



3- Descripción

Elemento que referencia las definiciones de los valores de cualquier atributo “[role](#)” usados en los elementos enlace “[footnote](#)” o “[linkbaseRef](#)” y que deben aparecer inmediatamente después de los posibles elementos “linkbaseRef”.

4- Ejemplo [27]:

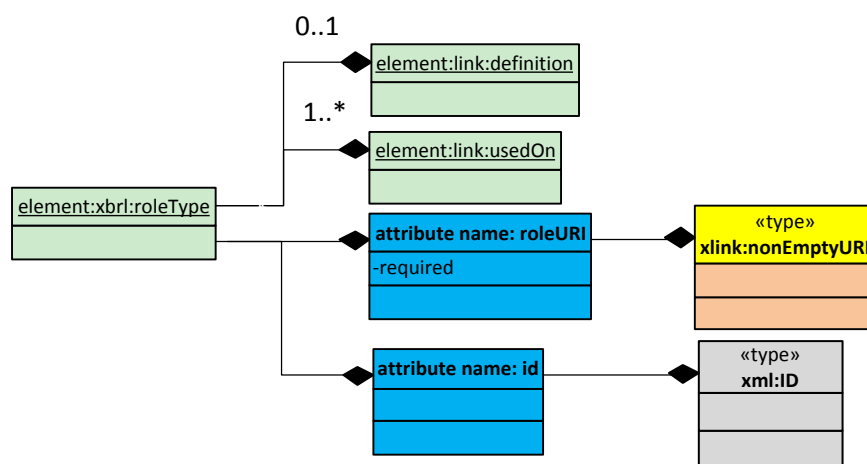
```
<link:roleRef roleURI="http://www.bde.es/es/fr/esrs/finrep/6-2008/2008-11-26/role/BalancePublicoConsolidado"
xlink:type="simple" xlink:href="es-be-FINREP-rol.xsd#BalancePublicoConsolidado"/>
```

Elemento link:roleType

1- Definición

```
<element name="roleType">
  <annotation>
    <documentation>
      The roleType element definition - used to define custom
      role values in XBRL extended links.
    </documentation>
  </annotation>
  <complexType>
    <sequence>
      <element ref="link:definition" minOccurs="0"/>
      <element ref="link:usedOn" maxOccurs="unbounded"/>
    </sequence>
    <attribute name="roleURI" type="xlink:nonEmptyURI" use="required"/>
    <attribute name="id" type="ID"/>
  </complexType>
</element>
```

2- Modelo



3- Descripción

Elemento que declara los valores del atributo “[role](#)” en un elemento enlace en una aplicación.

4- Ejemplo [27]:

```
<link:roleType roleURI="http://www.eurofiling.info/dsi/ct_17" id="dsict_17">
```



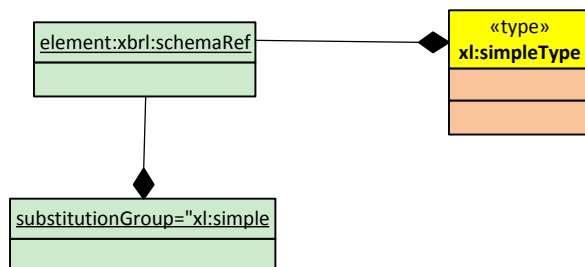
```
<link:definition>Equity and debt instruments held, and derivatives by original currency and  
counterparty residence (Part 1)</link:definition>  
<link:usedOn>link:definitionLink</link:usedOn>  
</link:roleType>
```

Elemento *link:schemaRef*

1- Definición

```
<element name="schemaRef" type="xl:simpleType" substitutionGroup="xl:simple">  
  <annotation>  
    <documentation>  
      Definition of the schemaRef element - used  
      to link to XBRL taxonomy schemas from  
      XBRL instances.  
    </documentation>  
  </annotation>  
</element>
```

2- Modelo



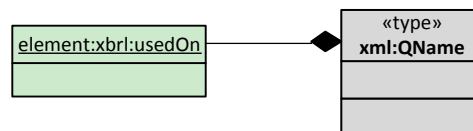
3- Descripción

Elemento para que los documentos instancia referencien a los esquemas taxonómicos.

1- Definición

```
<element name="usedOn" type="QName">  
  <annotation>  
    <documentation>  
      Definition of the usedOn element - used  
      to identify what elements may use a  
      taxonomy defined role or arc role value.  
    </documentation>  
  </annotation>  
</element>
```

2- Modelo



3- Descripción

Elemento que identifica los valores posibles de una taxonomía para los atributos [“role”](#) o [“arcrole”](#).

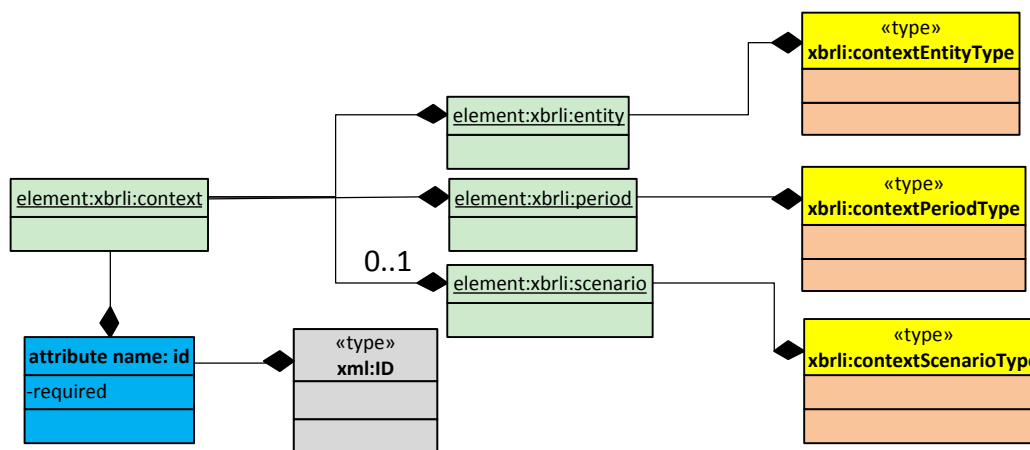
1- Definición

```

<element name="context">
  <annotation>
    <documentation>
      Used for an island of context to which facts can be related.
    </documentation>
  </annotation>
  <complexType>
    <sequence>
      <element name="entity" type="xbrli:contextEntityType" />
      <element name="period" type="xbrli:contextPeriodType" />
      <element name="scenario" type="xbrli:contextScenarioType" minOccurs="0" />
    </sequence>
    <attribute name="id" type="ID" use="required" />
  </complexType>
</element>

```

2- Modelo



3- Descripción

El elemento context es uno de los objetos principales en XBRL, debido a su contenido semántico. Define un conjunto de dimensiones que sirven para situar un hecho concreto. Está compuesto por una dimensión [Entidad](#), que puede tener de “0 a 1” [segmentos](#), es decir, de “0 a 1” dimensiones de acuerdo con la lógica de negocio. Además, deberá tener una dimensión [periodo de tiempo](#). Por último, se podrá definir un conjunto de dimensiones que definen un escenario.

A continuación una explicación gráfica del elemento “context”, en la cual podemos ver como la tabla de hechos es el propio elemento definido, y tendrá las dimensiones “Entity”, “Period”, “Segment” y “Scenariio”. Esta última podrá estar formada a su vez por varias dimensiones (“Scenariio1”,

“Scenario2”...“ScenarioX”). Todas esas dimensiones definirán un hecho en propia tabla de hechos (Figura 11):

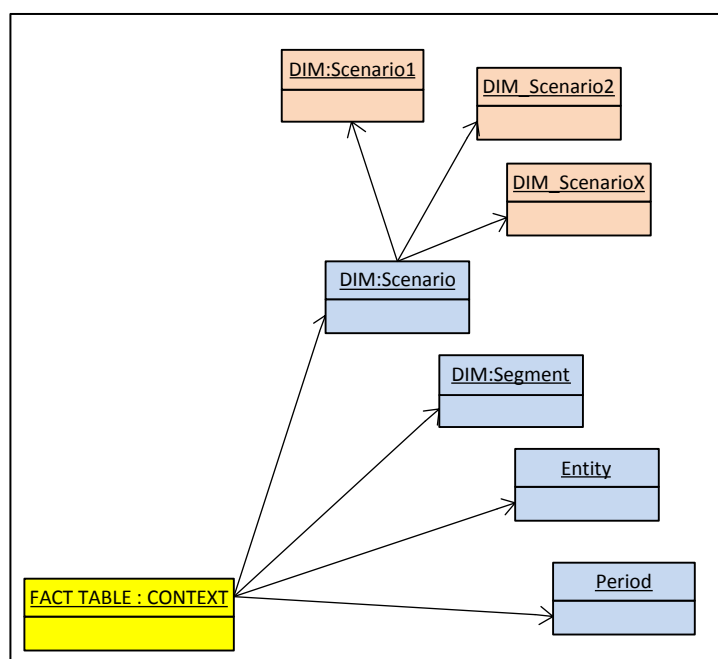


Figura 11. Elemento “context”

4- Ejemplo [27]:

```
<xbrli:context id="Context_Instant_GrupoConsolidableEntidadesCredito">
  <xbrli:entity>
    <xbrli:identifier scheme="http://www.ecb.int/stats/money/mfi">ES9000</xbrli:identifier>
    <xbrli:segment>
      <xbrldi:explicitMember dimension="es-be-cm-dim:Agrupacion">es-be-cm-
dim:AgrupacionIndividual</xbrldi:explicitMember>
    </xbrli:segment>
  </xbrli:entity>
  <xbrli:period>
    <xbrli:instant>2008-09-30</xbrli:instant>
  </xbrli:period>
  <xbrli:scenario>
    <xbrldi:explicitMember dimension="es-be-d-FR-dist:DistribucionDimension">es-be-d-FR-
dist:GrupoConsolidableEntidadesCredito</xbrldi:explicitMember>
  </xbrli:scenario>
</xbrli:context>
```

El elemento context, para este ejemplo, contiene la dimensión “Entity”, en este caso “http://www.ecb.int/stats/money/mfi”, con el atributo de dimensión “ES9000”, la dimensión “Segment” es “es-be-cm-dim:Agrupacion”, y el atributo de dimensión es “es-be-cm-dim:AgrupacionIndividual” (la lógica de

negocio indica que está agrupado individualmente). Por último, la dimensión de periodo de tiempo, instantáneo, con un atributo de dimensión que es “2008-09-30”.

Imaginémonoslo con hechos concretos, donde la dimensión “Entity_mfi” tiene censadas las entidades que correspondan, la dimensión “SEGMENT_es-be-cm-dim:Agrupacion”, los segmentos, la dimensión “Period_instant” tiene los instantes y la dimensión “Scenario_1_es-be-d-FR-dist:DistribucionDimension” define los escenarios posibles. No hay que olvidar que también están los “Conceptos Básicos”, que indican el concepto al que hace referencia el hecho en sí, pero que no forma parte del concepto. Por último, tendremos la tabla de hechos, donde se definirán los propios hechos, cada registro tomará un valor, aquí nombrado como “FACT”, que será, en este caso, 50, y que tendrá sentido situándolo en un contexto con la entidad “ES9000”, el segmento “es-be-cm-dim:AgrupacionIndividual”, el periodo instantáneo “2008-09-30” y el escenario “es-be-d-FR-dist:GrupoConsolidableEntidadesCredito”, para el Concepto Básico “es-be-p-FINREP:PromemoriaRiesgosContingentes” (Figura 12).

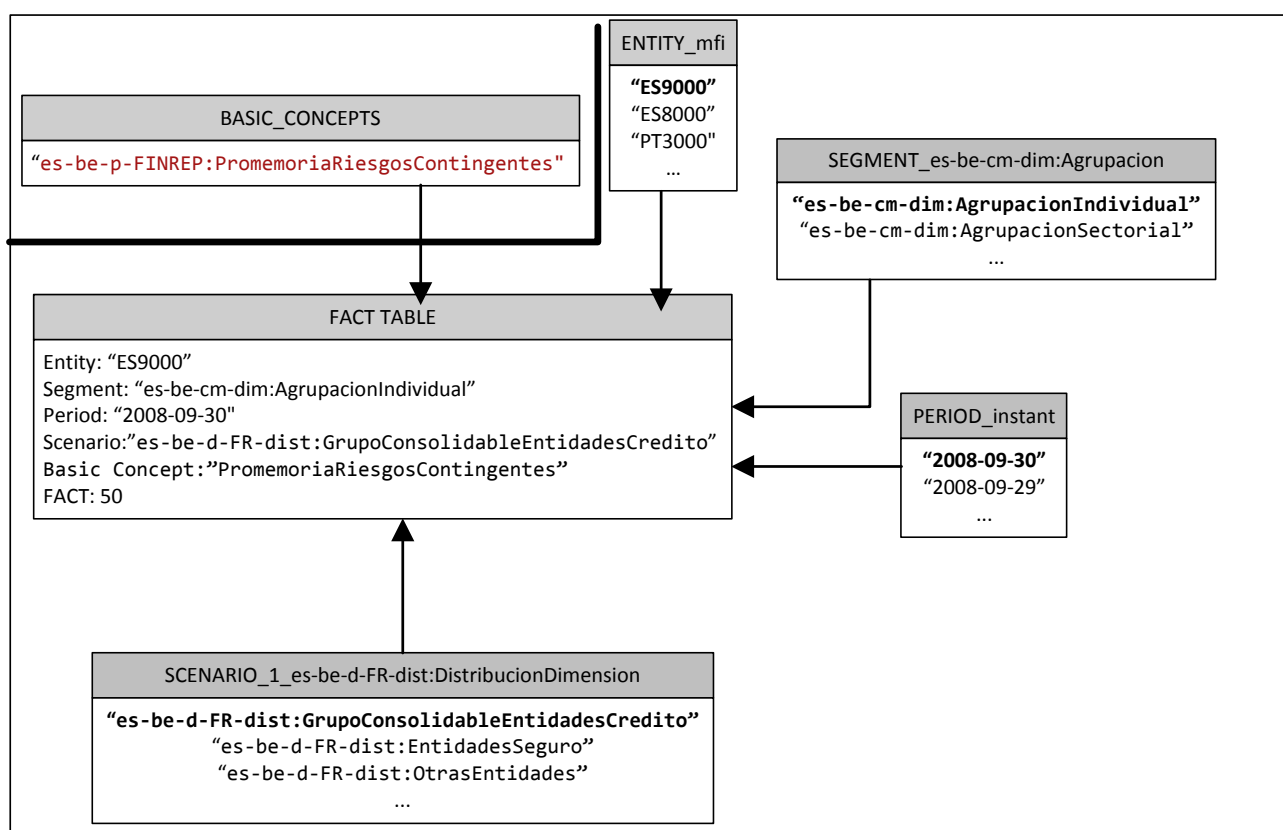


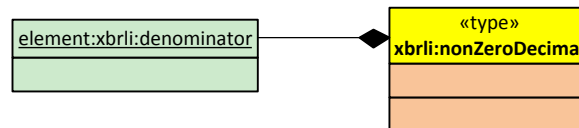
Figura 12. Ejemplo de modelo multidimensional para el elemento “context”

Elemento *xbrli:denominator*

1- Definición

```
<element name="denominator" type="xbrli:nonZeroDecimal" />
```

2- Modelo



3- Descripción

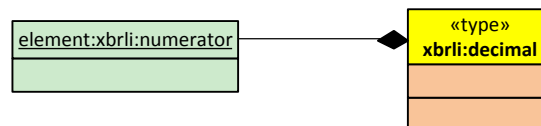
Elemento destinado a ser el denominador de una [fracción](#).

Elemento *xbrli:numerator*

1- Definición

```
<element name="numerator" type="decimal" />
```

2- Modelo



3- Descripción

Elemento destinado a ser el numerador de una [fracción](#).

Elemento *xbrli:divide*

1- Definición

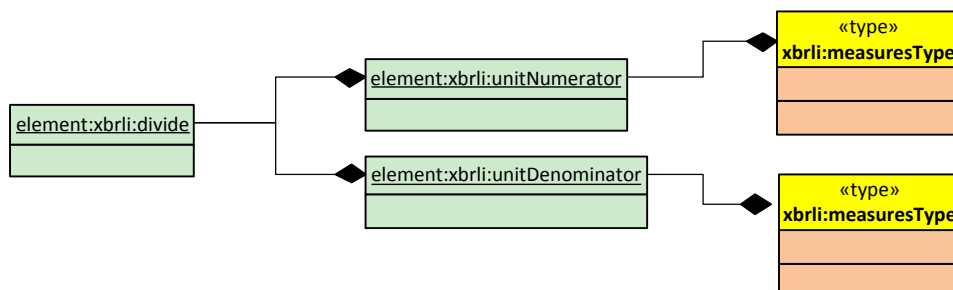
```
<element name="divide">
  <annotation>
    <documentation>
      Element used to represent division in units
    </documentation>
  </annotation>
  <complexType>
    <sequence>
      <element name="unitNumerator" type="xbrli:measuresType" />
      <element name="unitDenominator" type="xbrli:measuresType" />
    </sequence>
  </complexType>
</element>
```

```

    </sequence>
  </complexType>
</element>

```

2- Modelo



3- Descripción

Elemento para expresar un cociente, conteniendo obligatoriamente un [numerador](#) y un [denominador](#), acompañados ambos de un tip “measureType”.

Elemento *xbrli:item*

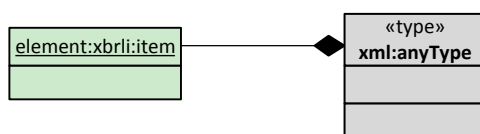
1- Definición

```

<element name="item" type="anyType" abstract="true">
  <annotation>
    <documentation>
      Abstract item element used as head of item substitution group
    </documentation>
  </annotation>
</element>

```

2- Modelo



3- Descripción

Definición de elemento simple.

Elemento *xbrli:measure*

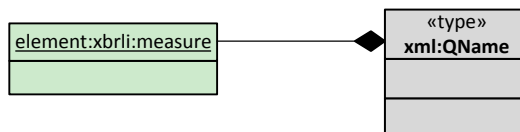
1- Definición

```

<element name="measure" type="QName" />

```

2- Modelo



3- Descripción

Elemento medida que define unidades simples.

1- Definición

```
<element name="segment">
  <complexType>
    <sequence>
      <any namespace="##other" processContents="lax"
        minOccurs="1" maxOccurs="unbounded" />
    </sequence>
  </complexType>
</element>
```

2- Modelo



3- Descripción

Elemento que contiene el segmento de negocios, o lógica de negocio, complementando aquellos casos en los que la [entidad](#) no sea suficiente.

4- Ejemplo [27]:

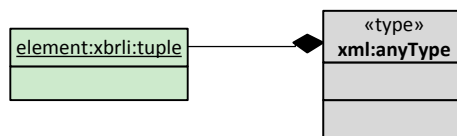
```
<xbrli:segment>
  <xbrldi:explicitMember dimension="es-be-cm-dim:Agrupacion">es-be-cm-
dim:AgrupacionIndividual</xbrldi:explicitMember>
</xbrli:segment>
```

En este caso la dimensión segmento es “es-be-cm-dim:Agrupacion”, y el atributo de dimensión es “es-be-cm-dim:AgrupacionIndividual”.

1- Definición

```
<element name="tuple" type="anyType" abstract="true">  
  <annotation>  
    <documentation>  
      Abstract tuple element used as head of tuple substitution group  
    </documentation>  
  </annotation>  
</element>
```

2- Modelo



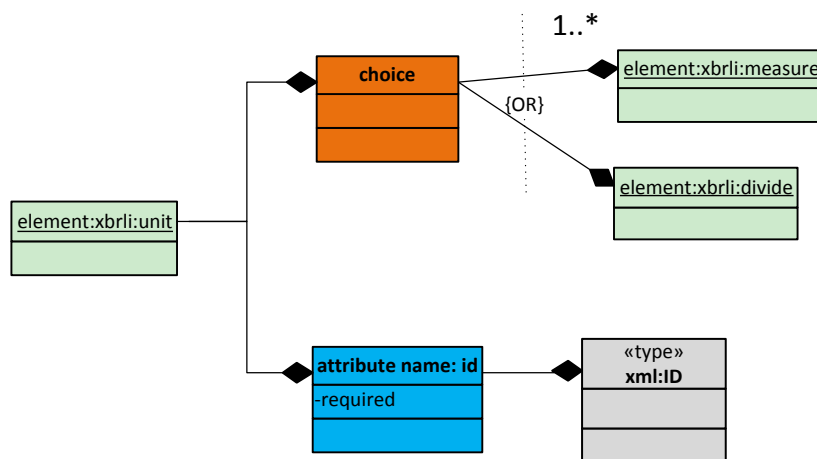
3- Descripción

Definición de elemento array.

1- Definición

```
<element name="unit">
  <annotation>
    <documentation>
      Element used to represent units information about numeric items
    </documentation>
  </annotation>
  <complexType>
    <choice>
      <element ref="xbrli:measure" minOccurs="1" maxOccurs="unbounded" />
      <element ref="xbrli:divide" />
    </choice>
    <attribute name="id" type="ID" use="required" />
  </complexType>
</element>
```

2- Modelo



3- Descripción

Elemento que especifica la unidad con la que se ha medido un ítem numérico. Si es simple, se dará con el elemento “[measure](#)”, y si es un cociente, con el elemento “[divide](#)”.

4- Ejemplo:

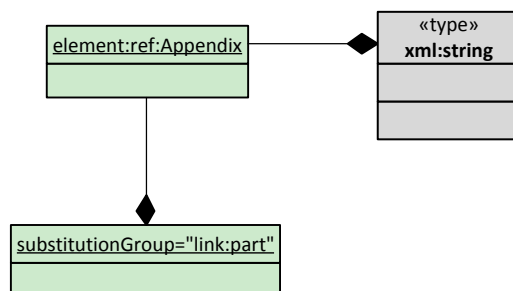
```
<xbrli:unit id="EURO">
  <xbrli:measure>iso4217:EUR</xbrli:measure>
</xbrli:unit>
```

Elemento ref:Appendix

1- Definición

```
<element name="Appendix" type="string" substitutionGroup="link:part" id="ref_Appendix">  
  <annotation>  
    <documentation xml:lang="en">Refers to the name of an Appendix, which could be a number or  
text.</documentation>  
  </annotation>  
</element>
```

2- Modelo



3- Descripción

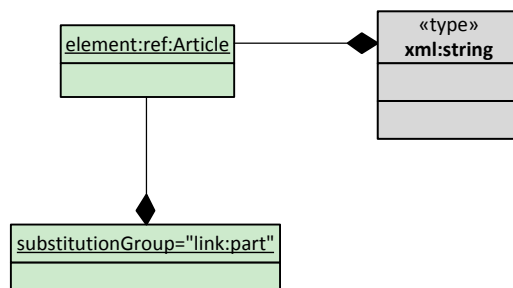
Explicación escrita de cómo está formado.

Elemento ref:Article

1- Definición

```
<element name="Article" type="string" substitutionGroup="link:part" id="ref_Article">  
  <annotation>  
    <documentation xml:lang="en">Article refers to a statutory article in legal  
material.</documentation>  
  </annotation>  
</element>
```

2- Modelo



3- Descripción

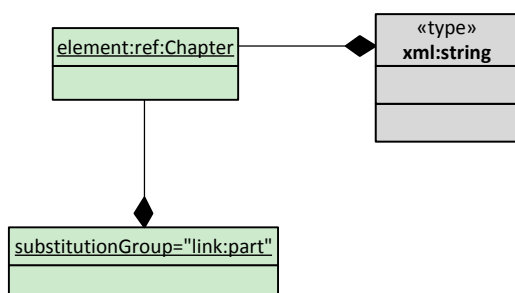
Elemento que contiene artículos adicionales.

Elemento ref:Chapter

1- Definición

```
<element name="Chapter" type="string" substitutionGroup="link:part" id="ref_Chapter">
  <annotation>
    <documentation xml:lang="en">For a publication that uses chapters, this part should be used to
capture this information. Because chapters are not necessarily numbers, this is a
string.</documentation>
  </annotation>
</element>
```

2- Modelo



3- Descripción

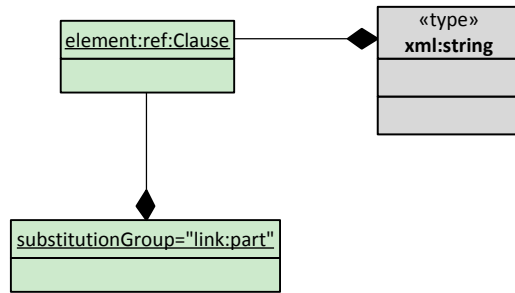
Elemento que indicará los capítulos de las publicaciones, si los hubiera.

Elemento ref:Clause

1- Definición

```
<element name="Clause" type="string" substitutionGroup="link:part" id="ref_Clause">
  <annotation>
    <documentation xml:lang="en">Sub component of a sub paragraph.</documentation>
  </annotation>
</element>
```

2- Modelo



3- Descripción

Elemento que contiene un componente de un elemento “[paragraph](#)”

Elemento *ref:Example*

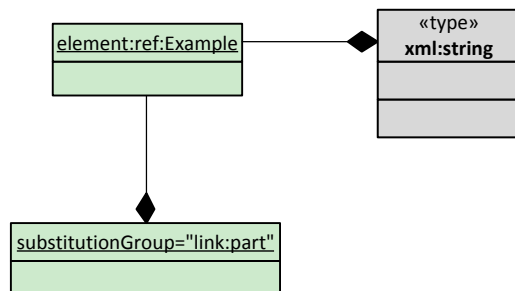
1- Definición

```

<element name="Example" type="string" substitutionGroup="link:part" id="ref_Example">
  <annotation>
    <documentation xml:lang="en">Example captures examples used in reference documentation;
there is a separate element for Exhibits.</documentation>
  </annotation>
</element>

```

2- Modelo



3- Descripción

Elemento que recoge ejemplos usados en la documentación referenciada.

Elemento *ref:Exhibit*

1- Definición

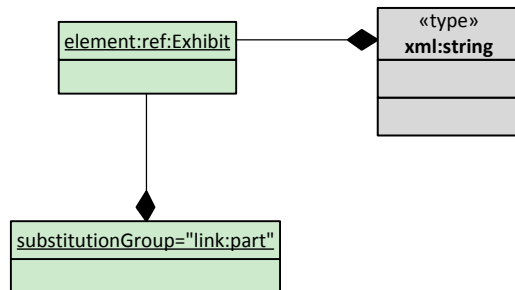
```

<element name="Exhibit" type="string" substitutionGroup="link:part" id="ref_Exhibit">
  <annotation>

```

`<documentation xml:lang="en">Exhibit refers to exhibits in reference documentation; examples have a separate element.</documentation>`
`</annotation>`
`</element>`

2- Modelo



3- Descripción

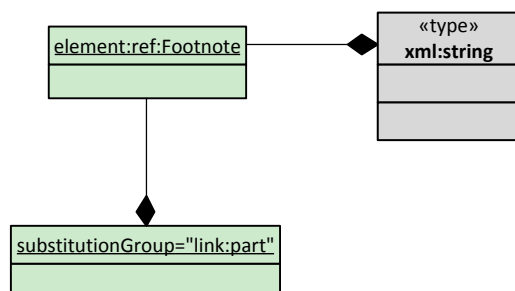
Elemento que recoge demostraciones en la documentación referenciada.

Elemento ref:Footnote

1- Definición

`<element name="Footnote" type="string" substitutionGroup="link:part" id="ref_Footnote">`
`<annotation>`
`<documentation xml:lang="en">Footnote is used to reference footnotes that appear in reference information.</documentation>`
`</annotation>`
`</element>`

2- Modelo



3- Descripción

Elemento que recoge referencias externas a textos legales. Un mismo “footnote” puede ser referenciado por diferentes objetos.

4- Ejemplo [14]:

```

<link:footnote>
  <link:loc xlink:type="locator" xlink:label="fact1" xlink:href="#item1"/>
  <link:footnoteArc
    xlink:type="arc "
    xlink:from="fact1" xlink:to="footnote 1 "
    xlink:title="ver nota aclaratori a "
    xlink:arcrole="http://www.xbrl.org/2003/arcrole/fact-footnote"/>

```

Elemento ref:IssueDate

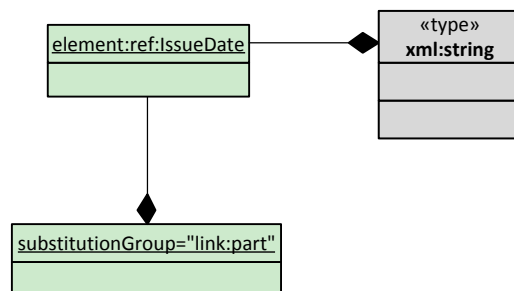
1- Definición

```

<element name="IssueDate" type="string" substitutionGroup="link:part" id="ref_IssueDate">
  <annotation>
    <documentation xml:lang="en">The issue date of the specific reference. The format is CCYY-
MM-DD.</documentation>
  </annotation>
</element>

```

2- Modelo



3- Descripción

Elemento que contiene la fecha específica de la referencia.

Elemento ref:Name

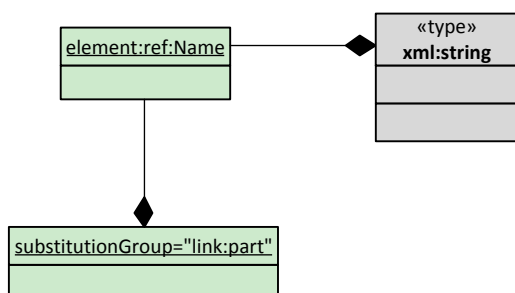
1- Definición

```

<element name="Name" type="string" substitutionGroup="link:part" id="ref_Name">
  <annotation>
    <documentation xml:lang="en">Name refers to the specific publication. For example,
"Statement of Financial Standards", "Statement of Position" or "IFRS". It does not include the
number.</documentation>
  </annotation>
</element>

```


2- Modelo



3- Descripción

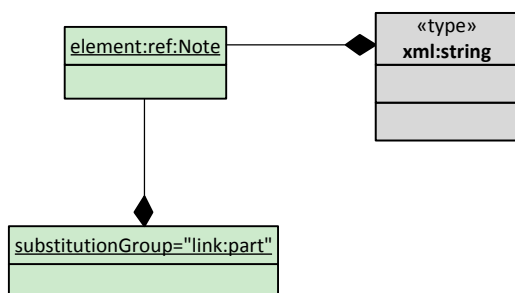
Elemento que contiene el nombre de la publicación específica.

Elemento *ref:Note*

1- Definición

```
<element name="Note" type="string" substitutionGroup="link:part" id="ref_Note">
  <annotation>
    <documentation xml:lang="en">Notes can contain reference material; use this element when the
note is published as a standalone document. There is a separate element for footnotes within other
references.
    </documentation>
  </annotation>
</element>
```

2- Modelo



3- Descripción

Elemento que contiene material referenciado.

Elemento *ref:Number*

1- Definición

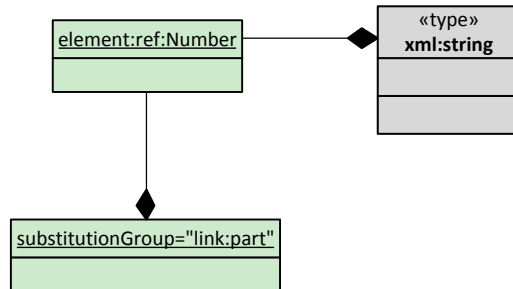
```
<element name="Number" type="string" substitutionGroup="link:part" id="ref_Number">
```

```

<annotation>
  <documentation xml:lang="en">Number is used to record the actual number of the specific
publication. For example, the number for FAS 133 would be 133.</documentation>
</annotation>
</element>

```

2- Modelo



3- Descripción

Elemento que recoge el número específico de la publicación.

Elemento ref:Page

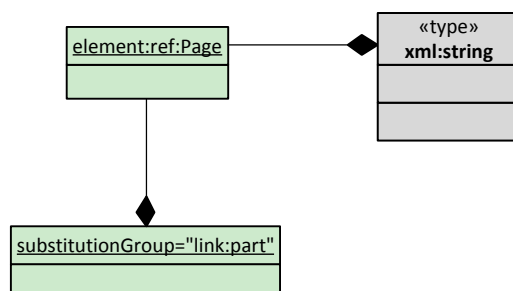
1- Definición

```

<element name="Page" type="string" substitutionGroup="link:part" id="ref_Page">
  <annotation>
    <documentation xml:lang="en">Page number of the reference material.</documentation>
  </annotation>
</element>

```

2- Modelo



3- Descripción

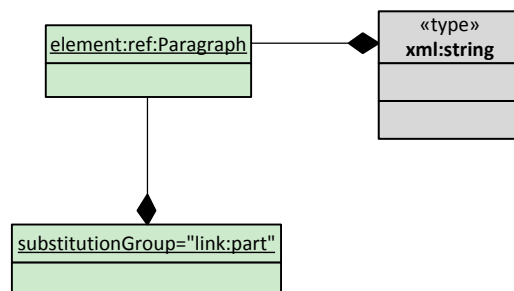
Elemento que recoge el número de página del material referenciado.

Elemento ref:Paragraph

1- Definición

```
<element name="Paragraph" type="string" substitutionGroup="link:part" id="ref_Paragraph">
  <annotation>
    <documentation xml:lang="en">Paragraph is used to refer to specific paragraphs in a
document.</documentation>
  </annotation>
</element>
```

2- Modelo



3- Descripción

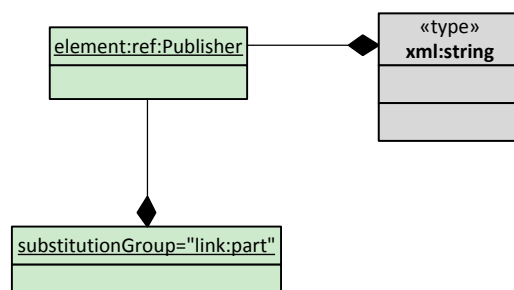
Elemento que indica el párrafo de un documento.

Elemento ref:Publisher

1- Definición

```
<element name="Publisher" type="string" substitutionGroup="link:part" id="ref_Publisher">
  <annotation>
    <documentation xml:lang="en">Publisher of the reference material, such as SEC, FASB, or
AICPA.</documentation>
  </annotation>
</element>
```

2- Modelo



3- Descripción

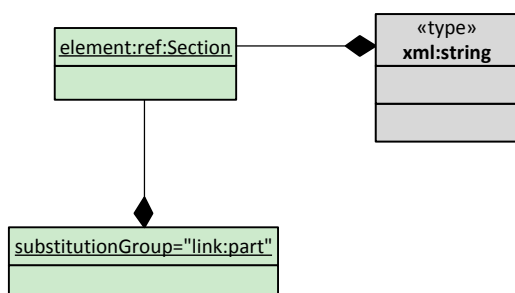
Elemento que indica el editor de una referencia.

Elemento ref:Section

1- Definición

```
<element name="Section" type="string" substitutionGroup="link:part" id="ref_Section">
  <annotation>
    <documentation xml:lang="en">Section is used to capture information typically captured in
sections of legislation or reference documents.</documentation>
  </annotation>
</element>
```

2- Modelo



3- Descripción

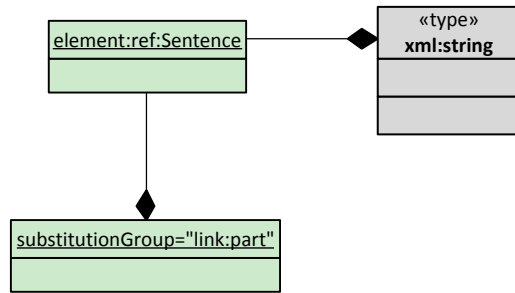
Elemento que recoge información habitual de documentos oficiales.

Elemento ref:Sentence

1- Definición

```
<element name="Sentence" type="string" substitutionGroup="link:part" id="ref_Sentence">
  <annotation>
    <documentation xml:lang="en">In some reference material individual sentences can be referred
to, and this element allows them to be referenced.</documentation>
  </annotation>
</element>
```

2- Modelo



3- Descripción

Elemento que contiene referencias exactas a citas.

Elemento ref:Subclause

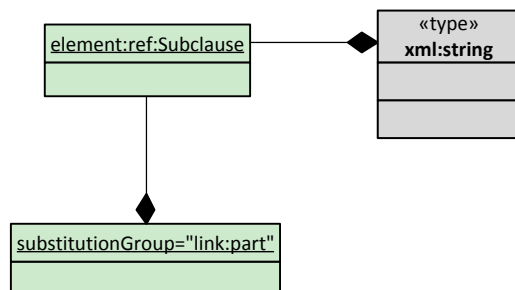
1- Definición

```

<element name="Subclause" type="string" substitutionGroup="link:part" id="ref_Subclause">
  <annotation>
    <documentation xml:lang="en">Subcomponent of a clause in a paragraph.</documentation>
  </annotation>
</element>

```

2- Modelo



3- Descripción

Elemento que recoge una subcláusula.

Elemento ref:Subparagraph

1- Definición

```

<element name="Subparagraph" type="string" substitutionGroup="link:part"
id="ref_Subparagraph">
  <annotation>
    <documentation xml:lang="en">Subparagraph of a paragraph.</documentation>
  </annotation>
</element>

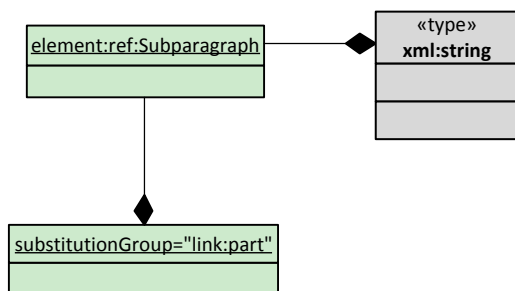
```

```

</annotation>
</element>

```

2- Modelo



3- Descripción

Elemento que recoge un subpárrafo.

Elemento ref:Subsection

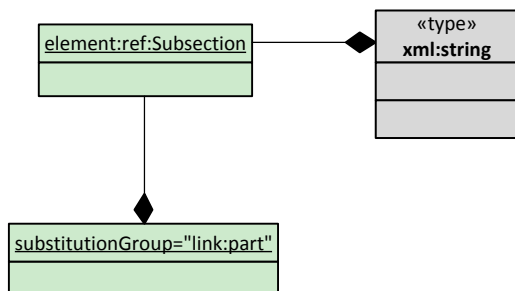
1- Definición

```

<element name="Subsection" type="string" substitutionGroup="link:part" id="ref_Subsection">
  <annotation>
    <documentation xml:lang="en">Subsection is a subsection of the section part.</documentation>
  </annotation>
</element>

```

2- Modelo



3- Descripción

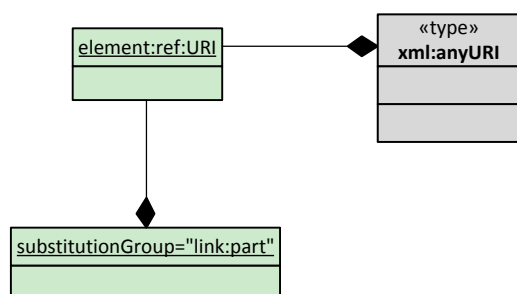
Elemento que recoge una subsección.

Elemento ref:URI

1- Definición

```
<element name="URI" type="anyURI" substitutionGroup="link:part" id="ref_URI">
  <annotation>
    <documentation xml:lang="en">Full URI of the reference such as
"http://www.fasb.org/fas133".</documentation>
  </annotation>
</element>
```

2- Modelo



3- Descripción

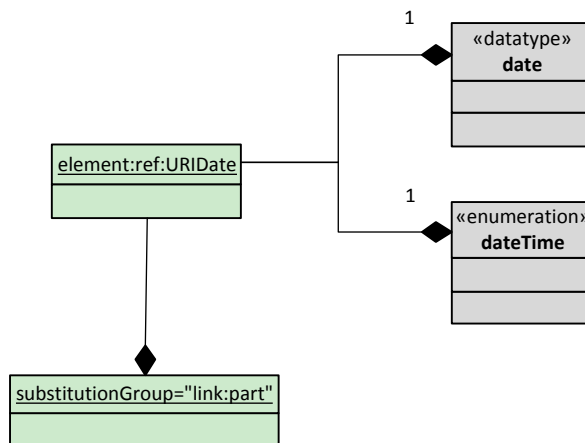
Elemento que recoge una URI completa.

Elemento ref:URIDate

1- Definición

```
<element name="URIDate" substitutionGroup="link:part" id="ref_URIDate">
  <annotation>
    <documentation xml:lang="en">Date or DateTime that the URI was valid, in CCYY-MM-DD
format.</documentation>
  </annotation>
  <simpleType>
    <union memberTypes="date dateTime" />
  </simpleType>
</element>
```

2- Modelo



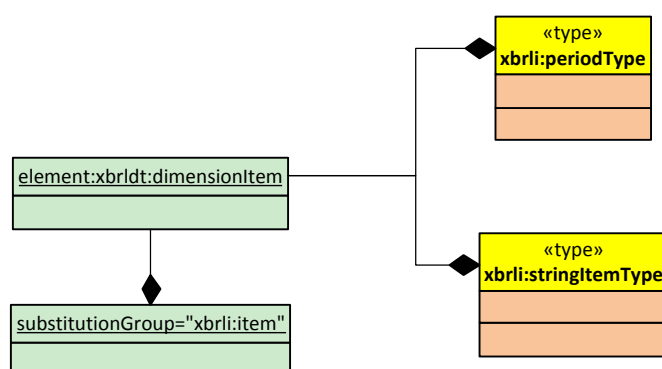
3- Descripción

Unión de los tipos de datos “date” y “dateTime”, para expresar la fecha de un URI.

1- Definición

```
<xs:element
  name="dimensionItem"
  id="xbrldt_dimensionItem"
  abstract="true"
  substitutionGroup="xbrli:item"
  type="xbrli:stringItemType"
  xbrli:periodType="duration"/>
```

2- Modelo



3- Descripción

Elemento que recoge conceptos multidimensionales en un marco predefinido.

Pueden ser:

Dimensiones explícitas (“explicit dimensions”)

Las dimensiones explícitas son aquellas en que los miembros del dominio pueden enumerarse explícitamente. Se representa por un elemento abstracto del grupo de sustitución de *xbrldt:dimensionItem* que no tiene atributo “[typedDomainRef](#)” y tiene arcos “dimension-domain” con cero o más miembros de dominio.

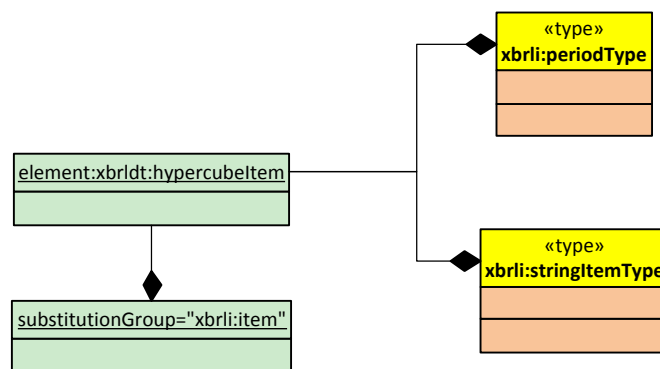
Dimensiones tipadas (typed dimensions)

Las dimensiones tipadas corresponden a aquellas en que el conjunto de miembros de su dominio es innumerable a nivel práctico. Se representa por un elemento abstracto del grupo de sustitución de *xbrldt:dimensionItem* que tiene un atributo “[typedDomainRef](#)”.

1- Definición

```
<xs:element  
  name="hypercubeltem"  
  id="xbrldt_hypercubeltem"  
  abstract="true"  
  substitutionGroup="xbrli:item"  
  type="xbrli:stringItemType"  
  xbrli:periodType="duration"/>
```

2- Modelo



3- Descripción

El elemento hipercubo en XBRL define una restricción que indica el número de dimensiones asociadas válidas o inválidas. Veamos un sencillo ejemplo:

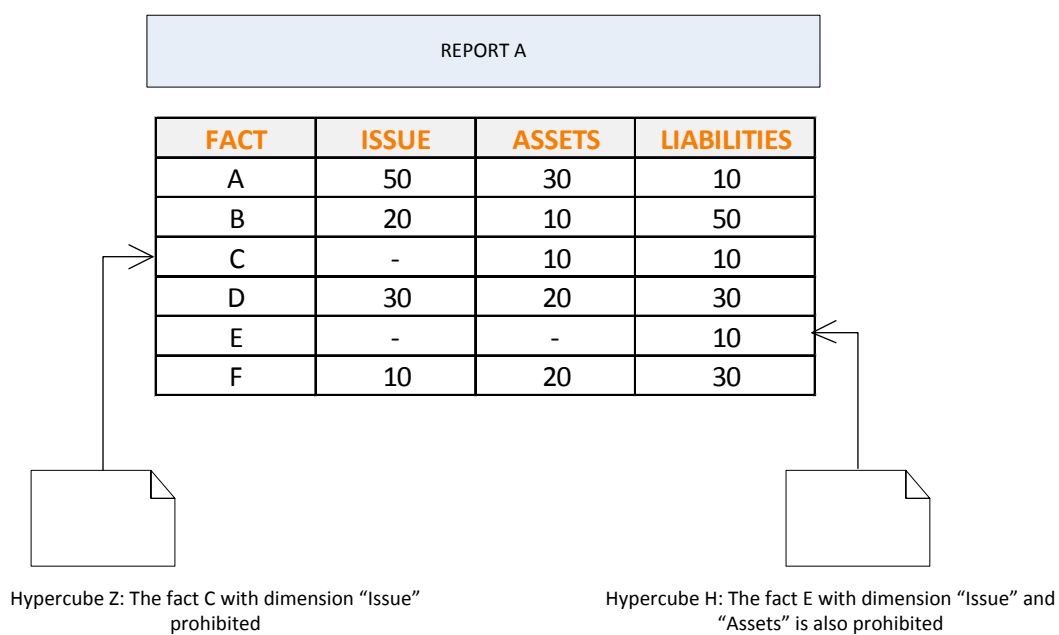


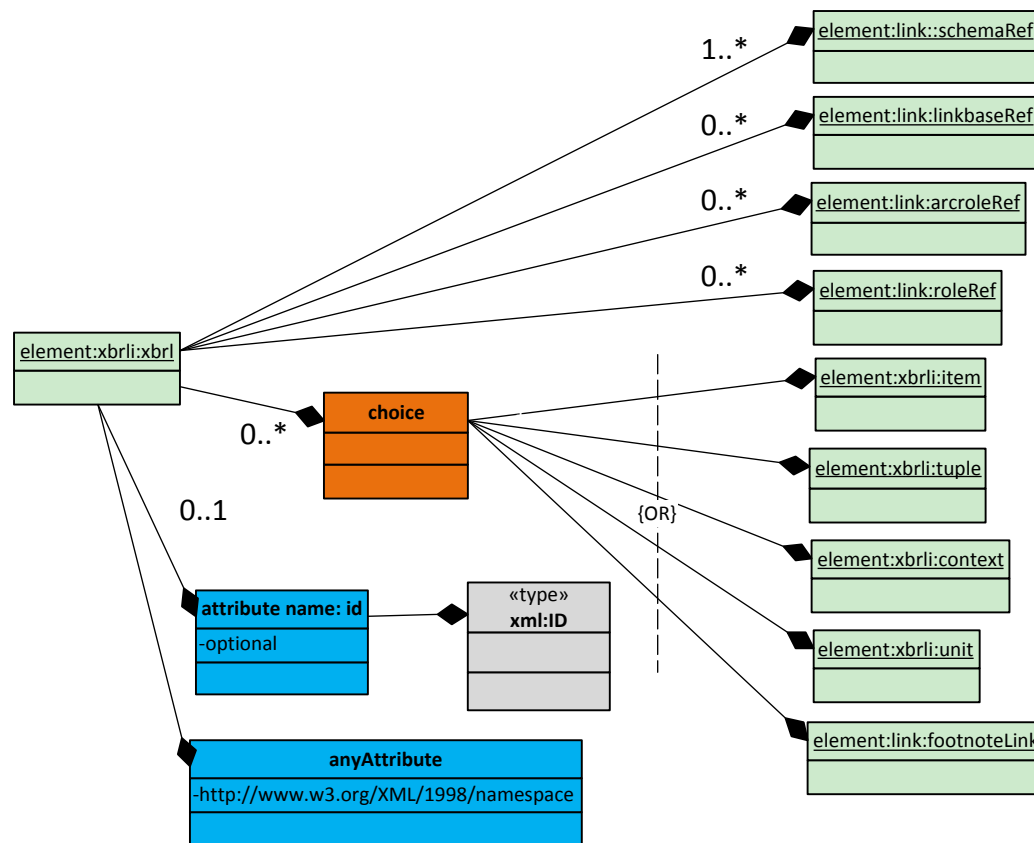
Figura 13. Ejemplo de elemento "hypercube"

Los hechos "C" y "E" tienen dos restricciones que indican, para "C", que el concepto "ISSUE" no se rellana, y que para "E", los conceptos "ISSUE" y "ASSETS", tampoco.

1- Definición

```
<element name="xbrl">
  <annotation>
    <documentation>
      XBRL instance root element.
    </documentation>
  </annotation>
  <complexType>
    <sequence>
      <element ref="link:schemaRef" minOccurs="1" maxOccurs="unbounded" />
      <element ref="link:linkbaseRef" minOccurs="0" maxOccurs="unbounded" />
      <element ref="link:roleRef" minOccurs="0" maxOccurs="unbounded" />
      <element ref="link:arcroleRef" minOccurs="0" maxOccurs="unbounded" />
      <choice minOccurs="0" maxOccurs="unbounded">
        <element ref="xbrli:item"/>
        <element ref="xbrli:tuple"/>
        <element ref="xbrli:context"/>
        <element ref="xbrli:unit"/>
        <element ref="link:footnoteLink"/>
      </choice>
    </sequence>
    <attribute name="id" type="ID" use="optional" />
    <anyAttribute namespace="http://www.w3.org/XML/1998/namespace" processContents="lax"
  />
</complexType>
</element>
```

2- Modelo



3- Descripción

Elemento raíz de todos los documentos instancia. Es el elemento contenedor de los datos en formato XBRL.

4- Ejemplo [27]:

```

<xbrli:xbrl>
  <link:schemaRef xlink:type="simple" xlink:href="http://www.bde.es/es/fr/esrs/finrep/6-
2008/2008-11-26/es-be-finrep-consolidado.xsd"/>
  <xbrli:context id="Context_Instant"></xbrli:context>
  <xbrli:context id="IS1"></xbrli:context>
  <xbrli:context id="Context_Instant_GrupoConsolidableEntidadesCredito"></xbrli:context>
  <xbrli:context id="Context_Instant_EntidadesSeguros"></xbrli:context>
  <xbrli:context id="pblo_FINREP"></xbrli:context>
  <xbrli:unit id="EURO"></xbrli:unit>
  <es-be-p-FINREP:PromemoriaRiesgosContingentes decimals="-3" contextRef="Context_Instant"
unitRef="EURO">0</es-be-p-FINREP:PromemoriaRiesgosContingentes>
  <ifrs-gp:CashAndBalancesWithCentralBanks decimals="-3" contextRef="Context_Instant"
unitRef="EURO">6316000</ifrs-gp:CashAndBalancesWithCentralBanks>
  <ifrs-gp:FinancialAssetsHeldForTradingTotal decimals="-3" contextRef="Context_Instant"
unitRef="EURO">25680000</ifrs-gp:FinancialAssetsHeldForTradingTotal>

```

```

<es-be-p-FINREP:CarteraNegociacionDepositoEntidadesCredito decimals="-3"
contextRef="Context_Instant" unitRef="EURO">4366000</es-be-p-
FINREP:CarteraNegociacionDepositoEntidadesCredito>
</xbrli:xbrl>

```

A continuación, y dado que son parte esencial del elemento “xbrl”, y de los propios meta-metadatos, mostramos otro ejemplo donde podemos observar la diferencia entre ítems y tuplas [29]:

Ítem:

```

<es-be-p-FINREP:CarteraNegociacionDepositoEntidadesCredito decimals="-3"
contextRef="Context_Instant" unitRef="EURO">4366000</es-be-p-
FINREP:CarteraNegociacionDepositoEntidadesCredito>

```

Tupla:

```

<pgc07mc-apdo0:PresentacionCuentasTupla>
  <pgc07mc-apdo0:YearFechaInicioCuentas contextRef="D.ACTUAL">2009</pgc07mc-
apdo0:YearFechaInicioCuentas>
  <pgc07mc-apdo0:MonthFechaInicioCuentas contextRef="D.ACTUAL">01</pgc07mc-
apdo0:MonthFechaInicioCuentas>
  <pgc07mc-apdo0:DayFechaInicioCuentas contextRef="D.ACTUAL">01</pgc07mc-
apdo0:DayFechaInicioCuentas>
  <pgc07mc-apdo0:YearFechaCierreCuentas contextRef="D.ACTUAL">2009</pgc07mc-
apdo0:YearFechaCierreCuentas>
  <pgc07mc-apdo0:MonthFechaCierreCuentas contextRef="D.ACTUAL">12</pgc07mc-
apdo0:MonthFechaCierreCuentas>
  <pgc07mc-apdo0:DayFechaCierreCuentas contextRef="D.ACTUAL">31</pgc07mc-
apdo0:DayFechaCierreCuentas>
  <dgi-dat-inf:TotalPagesPresented decimals="0" contextRef="D.ACTUAL" unitRef="pure">43</dgi-
dat-inf:TotalPagesPresented>
</pgc07mc-apdo0:PresentacionCuentasTupla>
<pgc07mc-apdo0:PresentacionCuentasTupla>
  <pgc07mc-apdo0:YearFechaInicioCuentas contextRef="D.ANTERIOR">2008</pgc07mc-
apdo0:YearFechaInicioCuentas>
  <pgc07mc-apdo0:MonthFechaInicioCuentas contextRef="D.ANTERIOR">01</pgc07mc-
apdo0:MonthFechaInicioCuentas>
  <pgc07mc-apdo0:DayFechaInicioCuentas contextRef="D.ANTERIOR">01</pgc07mc-
apdo0:DayFechaInicioCuentas>
  <pgc07mc-apdo0:YearFechaCierreCuentas contextRef="D.ANTERIOR">2008</pgc07mc-
apdo0:YearFechaCierreCuentas>
  <pgc07mc-apdo0:MonthFechaCierreCuentas contextRef="D.ANTERIOR">12</pgc07mc-
apdo0:MonthFechaCierreCuentas>
  <pgc07mc-apdo0:DayFechaCierreCuentas contextRef="D.ANTERIOR">31</pgc07mc-
apdo0:DayFechaCierreCuentas>

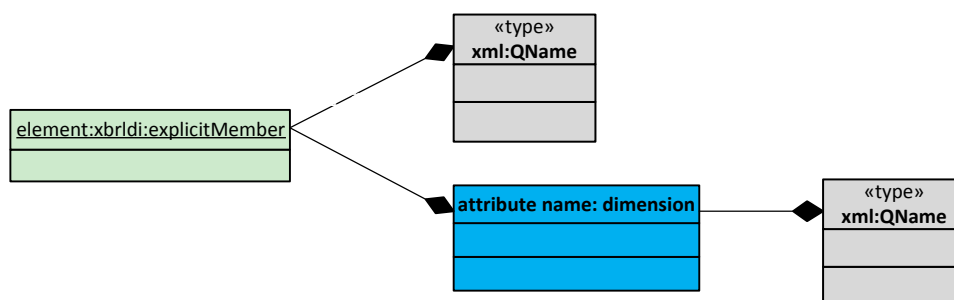
```

```
<dgi-dat-inf:TotalPagesPresented decimals="0" contextRef="D.ANTERIOR" unitRef="pure">0</dgi-  
dat-inf:TotalPagesPresented>  
</pgc07mc-apdo0:PresentacionCuentasTupla>
```

1- Definición

```
<element name="explicitMember">
  <annotation>
    <documentation xml:lang="en">This element contains the QName of an item that is
a member of an explicit dimension.
    </documentation>
  </annotation>
  <complexType>
    <simpleContent>
      <extension base="QName">
        <attribute name="dimension" type="QName" use="required"/>
      </extension>
    </simpleContent>
  </complexType>
</element>
```

2- Modelo



3- Descripción

Este elemento define una dimensión en la cual conocemos de antemano los atributos de dimensión.

4- Ejemplo [27]:

```
<xbrli:context id="Context_Instant_GrupoConsolidableEntidadesCredito">
  <xbrli:entity>
    <xbrli:identifier scheme="http://www.ecb.int/stats/money/mfi">ES9000</xbrli:identifier>
    <xbrli:segment>
      <xbrldi:explicitMember dimension="es-be-cm-dim:Agrupacion">es-be-cm-
dim:AgrupacionIndividual</xbrldi:explicitMember>
    </xbrli:segment>
  </xbrli:entity>
  <xbrli:period>
    <xbrli:instant>2008-09-30</xbrli:instant>
  </xbrli:period>
</xbrli:context>
```



```

<xbrli:scenario>
  <xbrldi:explicitMember dimension="es-be-d-FR-dist:DistribucionDimension">es-be-d-FR-
dist:GrupoConsolidableEntidadesCredito</xbrldi:explicitMember>
</xbrli:scenario>
</xbrli:context>

```

En el ejemplo, tenemos la dimensión “es-be-d-FR-dist:DistribucionDimension” que tendrá en su documentación correspondiente los atributos de dimensiones, entre los que se encontrará “es-be-d-FR-dist:GrupoConsolidableEntidadesCredito”. En otras palabras, este atributo de dimensión está censado dentro de un dominio definido en el documento correspondiente, es decir, en “IS1 6610” tendremos definido el atributo “es-be-d-FR-dist:GrupoConsolidableEntidadesCredito” dentro de la dimensión “es-be-d-FR-dist:DistribucionDimension”, entre otros valores. Veámoslo gráficamente:

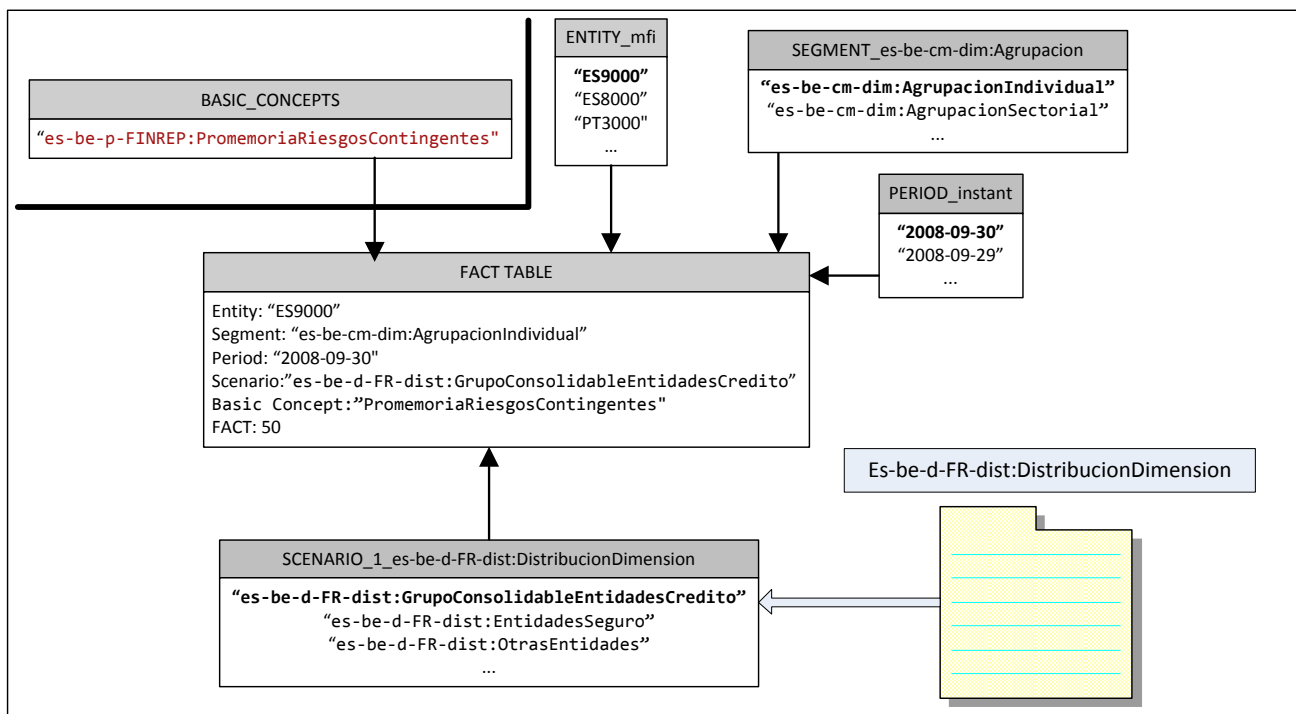
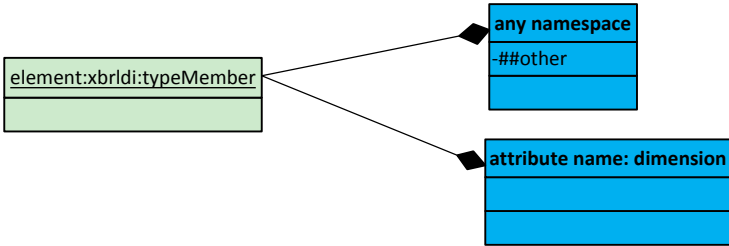


Figura 14. Ejemplo de elemento “explicitMember”

1- Definición

```
<element name="typedMember">
  <annotation>
    <documentation xml:lang="en">This element constains one child of anyType.
  </documentation>
  </annotation>
  <complexType>
    <sequence>
      <any namespace="##other"/>
    </sequence>
    <attribute name="dimension" type="QName" use="required"/>
  </complexType>
</element>
```

2- Modelo



3- Descripción

El elemento typeMember define una dimensión en la cual los atributos de dimensión no son conocidos. La dimensión existe, pero sin valores, que se cargarán cuando lo haga la instancia. Veámoslo gráficamente:

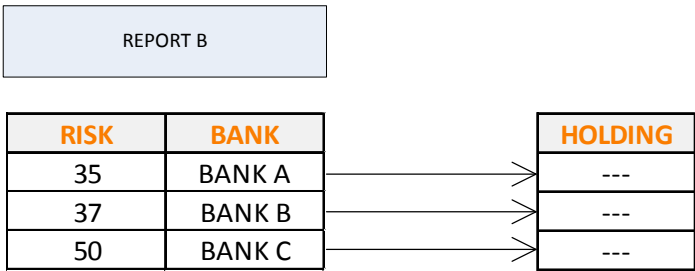


Figura 15. Ejemplo de “typeMember”

Los atributos de dimensión “BANK A”, “BANK B” y “BANK C” proceden de la dimensión “Holding”, pero ésta no está definida, es decir, los valores no forman parte de la taxonomía.

5. HERRAMIENTAS MODELADO XML-UML

A continuación vamos a analizar algunas de las herramientas que existen para modelar en UML. El objetivo es poder evaluar el coste, funcionalidad, y las ventajas y desventajas de cada producto, a la hora de llevar a UML distintos archivos XML, que son el origen de nuestra información.

Analizaremos productos libres y con varios sistemas de licenciamiento, en diferentes plataformas, y con diferentes funcionalidades. El objetivo de esto es poder recopilar información que nos lleve a tomar una decisión respecto a que herramientas utilizar para la generación de modelos de datos.

5.1 Herramientas

Microsoft Visual Studio 2010 Ultimate España

1- Descripción

Software para administración del ciclo de vida de las aplicaciones. Permite controlar desde el diseño hasta la implementación de la misma, tanto para nuevas aplicaciones como para mantenimiento de las ya desarrolladas.

2- Sistema de licenciamiento

Precio producto: Desde 12.769 Euros

Precio renovación de licencia: Desde 4.079 Euros.

3- Características

Manual de Usuario: Si

Soporte: Si

Importación de archivos XML: Si

Gratuita: No

Sistema Operativo: Windows

4- Dirección Web

<http://www.microsoft.com/spain/visualstudio/products/2010-editions/ultimate>

Altova

1- Descripción

Altova UModel sirve para establecer el punto inicial en el desarrollo de software. Permite modelar las aplicaciones en UML y generar código (Java, C#, Visual Basic .NET), así como documentación.

2- Sistema de licenciamiento

Precio producto:

Versión Enterprise: 199 Euros

Versión Professional: 109 Euros

3- Características

Manual de Usuario: Si

Soporte: Si

Importación de archivos XML: Si

Gratuita: No

Sistema Operativo: Windows

4- Dirección Web

<http://www.altova.com/>

Enterprise Architect

1- Descripción

Herramienta de análisis y diseño UML, participando en el desarrollo del software desde el análisis, pasando por los modelos de diseño, pruebas y mantenimiento. Multi-usuario, basada en Windows.

2- Sistema de licenciamiento

Edición Corporativa Flotante

Todas las características de la edición Corporativa, pero con sistema de licencias flotantes.

Precio compra:

Licencias 1 a 4: 335 dólares

Precio renovación de licencia (12 meses):

Licencias 1 a 4: 70 dólares

Edición Corporativa

Todas las características de la edición profesional. Posibilidad de utilizar diferentes gestores de Bases de Datos (SQL Server, Oracle, MySQL). Seguridad de usuario para modelos.

Precio compra:

Licencias 1 a 4 → 239 dólares

Precio renovación de licencia (12 meses):

Licencias 1 a 4 → 70 dólares

Edición Profesional

Modelado UML para analista, desarrollador y/o grupos de trabajo. Soporte para proyectos de múltiples usuarios. Importación, exportación y sincronización de código.

Precio compra:

Licencias 1 a 4 → 199 dólares

Precio renovación de licencias (12 meses):

Licencias 1 a 4 → 69 dólares

Edición Escritorio

Modelado UML para analista y/o desarrollador.

Precio compra:

Licencias 1 a 4 → 135 dólares

Precio renovación de licencias:

1 a 4 licencias: 45 dólares

*Existen descuentos especiales para estudiantes e instituciones educativas para la compra y renovación de licencias .

3- Características

Manual de Usuario: Si

Soporte: Si

Importación de archivos XML: Si

Gratuita: No

Sistema Operativo: Windows

4- Dirección Web

<http://www.sparxsystems.com/>

XML MODELING (hypermodel 3.1)

1- Descripción

Construido sobre la versión 3.2 de Eclipse y las especificaciones 2.1 de UML, permite gestionar grandes ficheros XML, así como diseñar nuevos proyectos UML o esquemas XML. Hypermodel 3.1 permite definir modelos de negocio y su integración con servicios Web.

Permite la importación desde XML, generación de XML Schema desde modelos UML, diseño de XML Schema, creación de diagramas UML y añade herramientas adicionales para diseño de UML y XML.

2- Sistema de licenciamiento

Software libre

3- Características

Manual de Usuario: Si

Soporte: Grupos de discusión, foros de usuarios.

Importación de archivos XML: Si

Gratuita: Si

Sistema Operativo: Linux, Mac

4- Dirección Web

<http://xmlmodeling.com/>

6. CONCLUSIÓN Y TRABAJO FUTURO

XBRL tiene como principal objetivo facilitar el intercambio de información financiera de una forma automática, definida semánticamente, y sintácticamente, y que además esta información es creada y validada en origen.

Dado que XBRL es un lenguaje con un fuerte contenido semántico, y hecho por el entorno financiero, es preciso estudiar tanto su sintaxis, analizando las reglas que lo forman, como su semántica, entendiendo el lenguaje XBRL, sus ideas, sus conceptos y sus definiciones.

Precisamente el ser un lenguaje realizado por economistas, con sus propias reglas, provoca que su complejidad sea mayor desde el punto de vista de la ingeniería informática. Conceptos que en un mundo tienen un significado y una implementación, en el otro pueden significar algo totalmente distinto (véase las diferencias, por ejemplo, en el concepto de hipercubo en el modelo de datos Multidimensional y el elemento “hypercube” en el modelo de datos XBRL).

El punto de partida de este trabajo es la falta de estudios de una definición del modelo de datos XBRL, y especialmente de los meta-metadatos. Este trabajo analizó los meta-metadatos haciendo una reingeniería inversa a partir de los esquemas XBRL, y sus roles.

Sin esta base es imposible conocer en toda su amplitud el lenguaje XBRL, y a su vez su modelo de datos. Este primer análisis es básico para realizar cualquier mapeo entre los diferentes modelos de datos XBRL (modelo multidimensional, modelo relacional) y con él se pretende una mejor comprensión de los niveles superiores, el metamodelo de datos y del modelo de datos XBRL. Estamos analizando el modelo, para a ser posible simplificarlo conociendo su estructura y sus reglas de negocio.

Este trabajo ayuda al desarrollador a conocer mejor el modelo de dato XBRL, de una forma fácil e intuitiva permitiéndole conocer los conceptos básicos del propio lenguaje.

Es precisamente esto último, el estudio de los niveles superiores, sobre lo que se fundamenta el trabajo futuro a desarrollar. El análisis y estudio del modelo de metadatos XBRL, formado por sus diferentes taxonomías, y, posteriormente, el propio modelo XBRL, para finalmente desarrollar técnicas de mapeo de este modelo de datos a otros, utilizando técnicas como MDA (“Model Driven Architecture”) [19], llegando a automatizar el proceso.

7. BIBLIOGRAFÍA

- [1] "A Gentle Introduction to SGML". C. M. Sperberg-McQueen and Lou Burnard. Digital Scholarship Services. University of Virginia Library, 2005.
- [2] AICPA, American Institute of CPAs. (www.aicpa.org/Pages/Default.aspx).
- [3] Asociación XBRL España (www.xbrl.es).
- [4] Association of Spanish Property and Commercial Registrars. (www.registradores.org/principal/indexx.jsp?id_idioma=2&prefijo_idioma=ING).
- [5] "Business Reporting in the New Economy" , (www.xbrl.org). J. Louis Matherne, Director – Information Technology, AICPA. Federation of Schools of Accountancy - 2000 Annual Meeting. December 4 and 5 - San Antonio, Texas.
- [6] Castro, E. 2004. "Una aproximación a la reutilización de estructuras documentales definidas con XML, en el marco de la web semántica". Thesis PHD. Universidad Carlos III de Madrid.
- [7] Desarrollo de Aplicaciones e Integración de Arquitecturas Sistemas sobre software libre y de código abierto en español (<http://xml.osmosislatina.com/>).
- [8] Diccionario de conceptos, términos y acrónimos relacionados con XBRL. Ricardo Correa, Daniella Caldana, Daniel Santaelices, Omar Rebolledo. 2009. (http://www.iaichile.com/resources/Documents/DICCIONARIO_XBRL_-_IAIGC.pdf)
- [9] Entrevista con Charles Hoffman, Inventor de XBRL. Localización: Banca y finanzas: Revista profesional de gestión financiera, ISSN 1135-0652, Nº 96, 2004, págs. 8-11.
- [10] Eurofiling Framework Taxonomy Architecture. Versión 0.1. 2009-01-04 (www.eurofiling.info/finrepTaxonomy/taxonomy/EFTA_20091217.pdf).
- [11] Felden, C. 2007. Multidimensional XBRL. DUV Deutscher Universitäts-Verlag. Book title: New Dimensions of Business Reporting and XBRL, pages 191-209.
- [12] Fiforms XML Definition Project. XS3P Project (http://xml.fiforms.org/xs3p/examples/po.xsd.html#type_PurchaseOrderType).
- [13] IVI Technologies. Stylus Studio (<http://xsd.stylusstudio.com>).
- [14] Libro Blanco XBRL. XBRL España (www.xbrl.es/downloads/libros/Libro_Blanco.pdf). 2006
- [15] Martín Quetglás, G. 2006. Curso de XBRL. Pearson Prentice Hall.
- [16] Morrison, Michael. XML al descubierto. Michael Morrison Ed. Prentice Hall, 2000.
- [17] MSDN. Microsoft Developer Network Platforms (<http://msdn.microsoft.com>).
- [18] Normas Internacionales de Información financiera. Universidad de Santiago de Chile (www.niif-ifs.usach.cl).
- [19] Object Management Group (www.omg.org/mda).
- [20] "Opening remarks part21". Autores. Ignacio Boixo, Gilles Moguet y Wolfgang Strohbach. Fecha 19 de noviembre de 2009. Editor: www.eurofiling.info. "XI European Banking Supervisors XBRL Workshop, 16th. - 20th. November 2009, Vienna".
- [21] O'Reilly "XML.com". Using W3C XML Schema (www.xml.com).
- [22] PGC, Spanish Accounting General Plan ("Plan General Contable", PGC), November, 20th, 2007. (www.icac.meh.es/Documentos/PGC_2007.pdf).
- [23] Phillip Engel, Walter Hamscher, Geoffrey Shuetrim, David vun Kannon, y Hugh Wallis. 2 de Julio de 2008. Extensible Business Reporting Language (XBRL) 2.1". XBRL International.
- [24] Real Academia Española (www.rae.es)

- [25] Santos, I., Castro, E. May, 2011. XBRL and the Multidimensional Data Model. WEBIST 2011, May, 6th-9th, Noordwijkerhout, The Netherlands
- [26] Schema Central (www.schemacentral.com/sc/xbml21/).
- [27] Taxonomía FINREP 2012.
- [28] Taxonomía IS1 6610.
- [29] Taxonomía Plan General Contable (PGC) 2011.
- [30] UN/CEFACT, "United Nations Center for Trade Facilitation and Electronic Business". (<http://www.unece.org/cefact/>).
- [31] W3Schools (www.w3schools.com/xml/).
- [32] What Is XBRL? Charles Hoffman. May 2008.
- [33] World Wide Web Consortium (W3C) (www.w3.org/XML).
- [34] XBRL International (www.xbrl.org).
- [35] "XBRL: The New Financial Reporting Language Has Arrived". Mary E. Phillips and Robert G. Colvard. Tennessee CPA Journal, Junio 2007.
- [36] "XML Bible". Elliotte Rusty Harold. IDG Books Worldwide, Inc. Tercera Edición. 2001.
- [37] XML Linkin Language (XLink) Version 1.1. Steve DeRose, Eve Maler, David Orchard, Norman Walsh (www.w3.org/TR/2010/REC-xlink11-20100506/)
- [38] XML Pointer Language (XPointer) Steven DeRose, Ron Daniel Jr, Paul Grosso, Eve Maler, Jonathan Marsh, Norman Walsh. (www.w3.org/TR/xptr/)
- [39] XML Schema. Second Edition. David C. Fallside, Priscilla Walmsley. 2004-10-24
- [40] XML y Tecnologías Relacionadas: Introducción. Diego Sánchez Schenone (www.ibm.com). 2011-07-29.